TI REVUE
DAS MAGAZIN FÜR TI 99/4A & TI PC

Nr. 5/85 — 2. Quartal — DM 5,50/ÖS 46/SFR 5,50

SUPER: PRIVATE KLEINANZEIGEN KOSTENLOS!

20 SEITEN LISTINGS FÜR IHREN TI 99/4 A

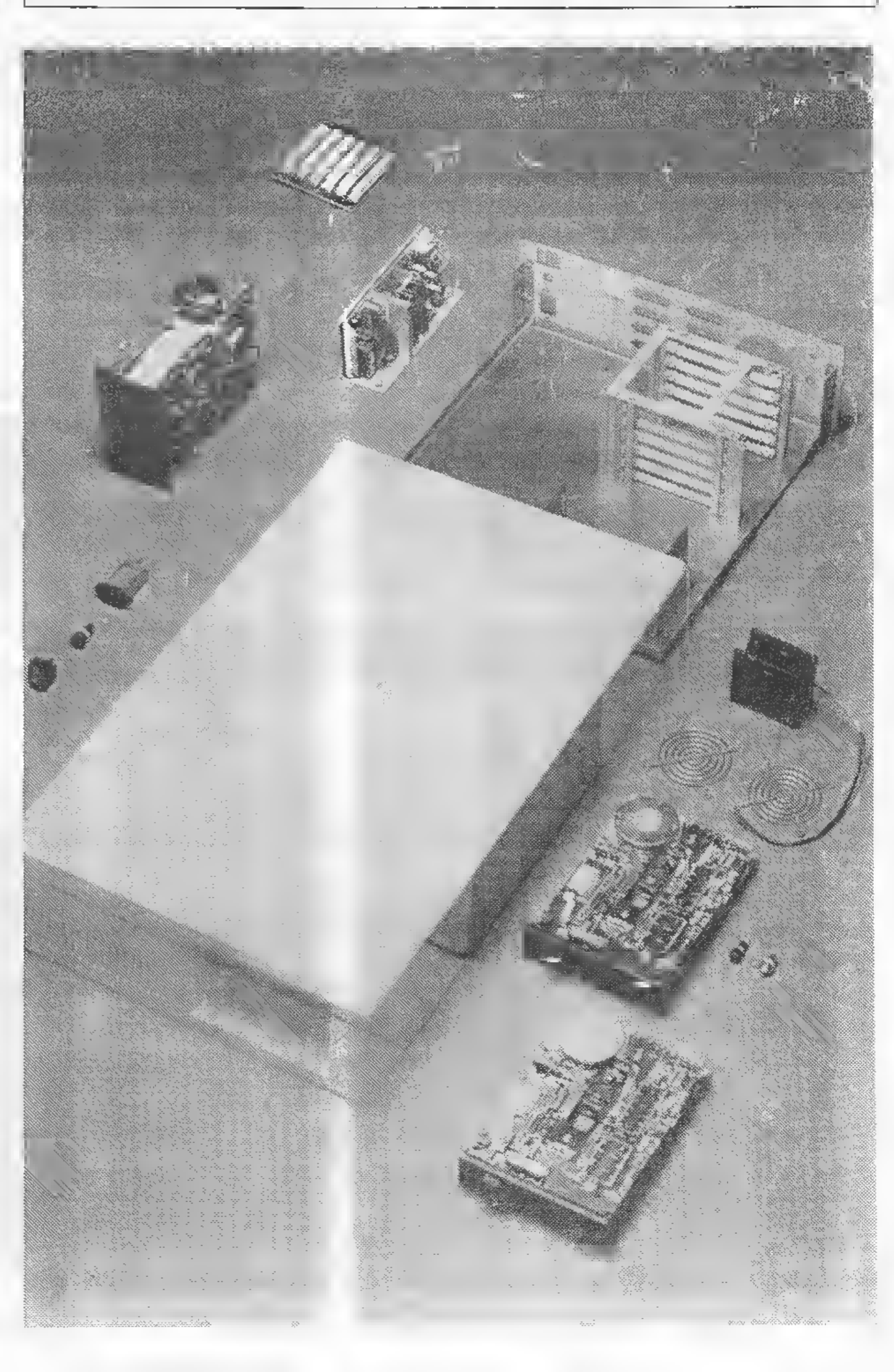
SERVICE TIPS & TRICKS

FERNSTEUERN MIT DEM TI

VERSTECKTE BEFEHLE IN TI MODULEN

NEU! ASSEMBLER KURS FÜR ANFÄNGER!

MACHEN SIE MEHR AUS IHREM 99er!



Grüß Gott–Gruezi–Guten Tag

Vor Ihnen liegt die 5. Ausgabe der TI-REVUE. Nun ist es ja gerade erst knapp über einen Monat her, da hatten wir die vierte Ausgabe ausgeliefert. Deshalb möchten wir hier nochmals erwähnen, daß die TI-REVUE normalerweise alle zwei Monate erscheint. Wir wollen momentan nur durch eine kürzere Erscheinungsweise die Verspätung, die Anfang des Jahres leider entstanden ist, wieder aufholen, so daß bis zum Jahresende wieder alles in der richtigen Bahn läuft.

Unser „heißes Telefon" für Eure Fragen entwickelt sich zu einem Erfolg. Unter der Nr. 0731/33220 stehen Euch jeden Dienstag von 15 bis 19 Uhr Experten für Eure Fragen zur Verfügung. Bitte nur Dienstags anrufen. An den anderen Wochentagen ist das Telefon nicht besetzt bzw. unser Redakteur, bei dem das heiße Telefon installiert ist, muß auch einmal etwas arbeiten (damit Ihr die TI-REVUE immer pünktlich in den Händen haltet). Achtung: Das heiße Telefon ist am Dienstag, den 19.6.85 und 25.6.85 nicht besetzt, *da ist besagter Redakteur im Urlaub (hat der es gut)!*

In den letzten Wochen hat der erweiterte Redaktionsstab auch alle bisher nicht beantworteten Anfragen aufgearbeitet. Hat einer von Euch bis heute noch keine Antwort auf einen Brief an uns, so müssen wir leider davon ausgehen, daß dieser Brief bei der Neuordnung verloren gegangen ist. So leid es uns tut, bitte schreibt uns Eure Fragen dann noch einmal oder meldet Euch über das heiße Telefon. Grundsätzlich bemühen wir uns, alle Eure Briefe und auch Eure Einsendungen von Programmen und Berichten innerhalb kürzester Zeit zu beantworten.

Einschließlich des Postwegs kann es aber schon bis zu drei Wochen dauern (in der Urlaubszeit vielleicht auch länger). Bei Berichten oder Programmen kann die endgültige Bestätigung auch etwas länger dauern, Ihr erhaltet dann aber vorab eine Eingangsbestätigung.

Das nächste Heft kommt Ende Juni, also nicht versäumen und bis dahin die besten Grüße von Eurem TI-REVUE-Team

Und nicht vergessen: Seid Ihr mit dem Heft unzufrieden, sagt es uns, Seid Ihr zufrieden, sagt es weiter.

Nr. 5/85
— 2. Quartal —

INHALT

IN DIESEM HEFT LESEN SIE

DIALOG

SERIE UND SERVICE

TEST, TIPS & TRICKS



Impressum

TI-REVUE, die Zeitschrift für den TI PC und den TI 99/4A erscheint 6×jährlich in der TI/CBM Verlagsgesellschaft, Postfach 1107, Elisabethstraße 1, 8044 Lohhof.
Redaktion: Senator-Presse-Service.
Mitarbeiter dieser Ausgabe: Udo Wenzel, Rudolf Trumbach, Volker Schirrmann, Torsten Rauer, Marcus Gröber, R. Bitter, Klaus Weidemann, Heiner Martin, Werner E. Seibt, Ugo Zoppi, Pieter Coates, Jürgen Heinzerling, Hartmut König.
Verantwortlich für den Inhalt: Klaus Herdin und Heiner Martin. Verantwortlich für Anzeigen: Bruno Redase.
Verantwortlich für Listings: Hartmut König.
Alle: Postfach 1107, 8044 Lohhof.
Anfragen bitte nur schriftlich.
Druck: Mayer und Söhne
Es gilt die Honorarliste des Verlages.

Gerichtsstand: München
Geschäftsführer: Werner E. Seibt
Abo- und Kassetten-Service: Henny Rose Seibt

LESERBRIEFE

BASICODE FÜR DEN TI 99/4A?

Könnten Sie mich informieren, ob es Basicode 2 für den TI 99/4A gibt?
Dirk Fischer
5757 Fröndenberg

Basicode ist eine besondere Form der Übermittlung von Basicprogrammen, die vom holländischen Radio Hilversum entwickelt worden ist und mittlerweile auch von WDR und NDR bei Computersendungen benutzt wird. Zwischenzeitlich werden diverse Programme in Basiccode auf Kassette angeboten. Grundgedanke dieser Entwicklung war es, die Programme von den verschiedenen Computertypen unabhängig zu machen. So befinden sich auf den Kassetten vor dem Programm jeweils eine spezielle Laderoutine für den besonderen Computertyp. Bisher ist es uns nicht bekannt, daß der Basiccode auch für den TI 99/4A zu haben ist, aber vielleicht kann hier einer der Leser weiterhelfen.

ADVENTURE-SPIELE AUF DISKETTE?

Meine Adventure-Spiele liegen auf Kassette vor und jeder Versuch, sie auf Diskette zu überspielen, schlug fehl. Am Anfang sah es zwar so aus, als könnte ich das Band mit Basic einlesen, aber nach "Daten ok, press enter" stieg der Rechner mit einer Fehlermeldung aus und es scheint auch kein Programm im Speicher abgelegt gewesen zu sein, da der Debugger in den relevanten Speichern nichts anzeigte.
Können Sie mir einen Weg schildern, auf dem ich es schaffe, Adventure-Spiele von Kassette auf Diskette zu übertragen? Das Einlesen von Diskette ist einfach viel schneller und komfortabler; außerdem habe ich normalerweise keinen Kassettenrekorder angeschlossen. Bei meinen Versuchen, einen Weg zu finden, stolperte ich noch über folgende Frage: Die Adventure-Spiele sind anscheinend wie ganz normale Basic-Programme gespeichert. Nur, in welcher Form werden diese eigentlich gespeichert? Mit dem TI-Writer und dem Assembler lassen sie sich jedenfalls nicht einlesen, da diese nie Fixed/Variable 80 Format akzeptieren.
Uwe Schulze
8501 Eckental-Forth

Wie Sie richtig bemerkt haben, sind die Adventure-Spiele in dem gleichen Format wie Basic-Programme abgespeichert, das sogenannte Programm-Format. Bei diesem Format wird der Inhalt des VDP-RAM-Speichers einfach auf das Speichermedium übergeben, die Anzahl Bytes wird vorher in dem zum Zugriff auf das Speichermedium notwendigen PAB angegeben. (Siehe hierzu die entsprechenden Ausführungen im Handbuch zum Editor-Assembler.)
Mit einer kleinen Assembler-Routine, die einmal über GPLLNK das Spiel von der Kassette einliest und dann wieder über DSRLNK auf die Diskette speichert. Benutzen Sie dazu die OP-Codes für LOAD und SAVE in den PAB's.
Übrigens ist die Verwendung des GPLLNK im Hinblick auf den Zugriff der Kassettenroutine im Editor/Assembler-Handbuch falsch beschrieben, bitte halten Sie sich hier an die Ausführungen im Handbuch zum Mini-Memory. Zu kompliziert? Gut, wir werden im Zuge unserer Artikel zum Thema Assembler noch näher darauf eingehen.
Es gibt da aber für das Adventure-Modul noch etwas ganz Feines: Einen Adventure-Editor zum Selberschreiben von Spielen für dieses Modul oder Ändern der Spiele. Wer diesen Editor benutzen will, muß sich aber darüber im Klaren sein, daß er eine neue Programmiersprache, nämlich die im Adventure-Modul enthaltene, lernen muß. Mit diesem Programm lassen sich die Spiele ebenfalls von Kassette auf Diskette übertragen.
Pikanterweise wird dieses Programm momentan nur in den USA (Fa. Texcomp) angeboten, obwohl es hier in der Bundesrepublik geschrieben wurde. Die Autoren sind hier noch auf der Suche nach einem Vertrieb.

DISK-LAUFWERKE FÜR DEN TI 99/4A

Mehrere Zuschriften erreichten uns in Bezug auf zum TI 99/4A passende Diskettenlaufwerke. Die hier veröffentlichten Auszüge sind nur stellvertretend:

*

Ist es prinzipiell möglich, ein Disk-Laufwerk eines xbeliebigen Herstellers (z.B. BASF) ohne eingebauten Controller mit einem geeigneten Controller zum TI 99/4A kompatibel zu gestalten?
Thomas Külpmann
5800 Hagen

*

Ich will mir in nächster Zeit einen Disketten-Controller für meinen TI zulegen. Nun möchte ich gerne wissen, ob ich an diesen jedes beliebige Laufwerk anschließen kann, oder nur ganz bestimmte, die zu meinem TI passen?
Armin Braun
7292 Mitteltal

*

Ich möchte mit Euch über den Anschluß von Diskettenlaufwerken sprechen. Meine Hardware besteht aus Speichererweiterung, Box, Controller und einer Schnittstelle mit 2 RS232C und einem Centronicsanschluß. Ich habe nun eine Menge Fragen an Euch: Das Laufwerk BASF 6148 Slimline mit 1,6 MB unformatiert habe ich mir ausgesucht. Meine Fragen nun:
1. Kann ich zwei Laufwerke dieses Typs überhaupt anschließen?
2. Gibt es Schwierigkeiten beim Anschluß an eine RS 232C Schnittstelle?
3. Wo bekomme ich das Disk-Manager-Modul?
4. Wozu ist ein 3. oder 4. Laufwerk gut?
5. Das Laufwerk hat 77/ Spuren je Oberfläche. Bringt das irgendwelche Probleme?
6. In den techn. Daten steht, daß die Lebensdauer 5 Jahre beträgt. Heißt das, daß ich das Laufwerk nach 5 Jahren verschrotten kann?
Walter Haidinger
A-3108 St. Pölten

Zum Anschluß eines Diskettenlaufwerkes an den TI wird zuallererst einmal ein Disk-Controller benötigt. Zu dessen Lieferumfang gehört auch ein Diskettenmanager. Der Anschluß eines Laufwerkes direkt an die RS 232C ist nicht möglich. Disk-Controller sind momentan in mehreren Ausführungen erhältlich: Der original TI-Controller für die P-Box, zwei Fremdfabrikate aus den USA ebenfalls für die P-Box und eine Mini-Erweiterung aus den USA, die mit einem Disk-Controller ausgerüstet ist bzw. werden kann. Allen diesen Controllern gemeinsam ist, daß die angeschlossenen Laufwerke 40 Spuren je Seite haben müssen. Sie können einseitig oder doppelseitig sein. Das ist also das erste Kriterium: 40 Spuren je Seite. Der nächste Punkt ist die Dichte, d.h. die Menge der Daten, die je Spur gespeichert werden können; der TI-Controller arbeitet nur mit einfacher Dichte, die neueren, aus den USA kommenden Controller erlauben auch das Arbeiten mit doppelter Dichte. Mit einfacher Dichte lassen sich 92K-Byte (184KByte bei doppelseitigen Laufwerken) speichern, bei doppelter Dichte sind es 184KByte (368KByte bei doppelseitigen Laufwerken). Die heute angebotenen Laufwerke eignen sich so gut wie alle für einfache (in den Datenblättern meist mit FM bezeichnet) und doppelte Dichte (MFM), so daß hier nur

bei Sonderangeboten aus irgendwelchen Restposten aufgepaßt werden muß. Nun muß noch die Schnittstelle des Laufwerkes zum Disk-Controller passen. Ein Laufwerk, welches mit dem 34poligen Platinensteckanschluß ausgerüstet ist, erfüllt im allgemeinen diese Bedingung. Fassen wir also nochmal zusammen: Wichtig ist ein 40 Spur (je Seite)-Laufwerk. Die Wahl, ob einseitig oder doppelseitig, ist mehr eine Frage des Preises. Ein weiterer wichtiger Punkt ist die Positionierungszeit bei einem Spurwechsel. Sie sollte beim gewählten Laufwerk bei 15 Millisekunden oder weniger liegen. Geeignete Diskettenlaufwerke gibt es unter anderem von den Herstellern ALPS, BASF, MPI, Shugart, TEAC und Toshiba, um nur einige zu nennen.

40-Spur-Laufwerke sind also das richtige für die Disk-Controller. Nun gibt es aber für den TI-Controller auch geänderte Betriebssysteme, bei denen auch 80-Spur-Laufwerke angeschlossen werden können. Hier geben dann aber die Vertriebsfirmen dieser Betriebssysteme Auskunft über die geeigneten Laufwerke. Laufwerke mit 77 Spuren sind aber in keinem Fall für den TI 99/4A geeignet.

Der Anschluß der Laufwerke erfolgt parallel an den Disk-Controller. Je nach Controller können maximal 3 oder 4 Laufwerke angeschlossen werden. Je mehr Laufwerke angeschlossen werden, desto mehr Speicherplatz auf den Disketten steht ohne Diskettenwechsel zur Verfügung. Die Anzahl der Laufwerke, die Sie Ihrem TI 99/4A spendieren wollen, ist also eine Frage der Bequemlichkeit, der zu verarbeitenden Datenmengen und schlußendlich eine Frage der Finanzen. Zu beachten ist noch, daß jedes Laufwerk auch noch eine Stromversorgung von 5V und 12V benötigt, die im Preis meist nicht enthalten ist.

Bleibt noch die Frage nach der Lebensdauer von Disk-Laufwerken. Die in den Datenblättern angegebene mittlere Lebensdauer bezieht sich auf eine professionelle Betriebszeit. Beim Einsatz mit Heimcomputern wird diese üblicherweise nicht täglich erreicht, und so liegt die Lebenserwartung durchaus höher. Ganz abgesehen davon, daß natürlich ein Laufwerk, wie jedes andere technische Produkt, einem gewissen Verschleiß unterliegt und derjenige, der es richtig pflegt, mit einer höheren Lebensdauer rechnen kann als jemand, der unpfleglich mit ihm umgeht.

UHRPROGRAMM AUCH OHNE SPEICHERERWEITERUNG?

Da nun einige Ausgaben der TI-REVUE auf dem Markt sind, möchte ich folgende Fragen an Sie stellen:
1. Im Heft Nr. 2 stellt Ihr ein Listing für eine Uhr im TI 99/4A vor. Für die Ausführung wird lediglich das Mini-Memory oder Ext.-Basic mit 32K Speichererweiterung benötigt. Nun möchte ich wissen, ob es auch ähnliche Programme für das Extended Basic ohne Speichererweiterung gibt?
2. Als ich das Heft Nr. 3 erhielt, tippte ich das Listing für den Klettermax ein. Doch obwohl es erst so aussah als würde es funktionieren, bemerkte ich, daß sich das Männchen zwar nach rechts und links, doch nicht nach oben und unten bewegen ließ. Ist in Ihre abgedruckten Listing ein Fehler?
3. Ich habe mit einigen Freunden einen Amateur-Video-Film gedreht und wollte nun wissen, ob man nicht mit dem TI 99/4A Schrifteinblendungen vornehmen kann. Vielleicht wissen Sie, was ich dazu brauche, um Bild und Computer-Darstellung gleichzeitig auf dem Fernseher erscheinen und mit dem Video-Recorder aufnehmen zu können.
Frank Hartlage
4650 Gelsenkirchen

Da ein Uhrenprogramm immer einen Zähler auf 0 setzen muß, welches mit dem Befehl CALL LOAD geschieht, ist eine Speichererweiterung beim Extended Basic leider unumgänglich. CALL LOAD funktioniert nur mit der Speichererweiterung.
In der Erklärung zum Spiel Klettermax hat sich ein Fehler eingeschlichen. Benutzen Sie die Tasten C und F, um das Männchen hoch und runter zu bewegen.
Bei Ihrer Frage zum Video-Film ist die Redaktion überfragt.
Wenn einer der Leser hier schon Erfahrungen gesammelt hat, so möchte er es uns bitte schreiben.

32K-SPEICHERERWEITERUNG DEFEKT!

Endlich ist sie da, die langersehnte 4. Ausgabe. Trotz aller Probleme („Tagebuch"), wieder ein gelungenes Werk!
– Zum Artikel Floppy: Ich habe ein Siemens-Laufwerk (550,– DM mit Zubehör) und es arbeitet 100%ig richtig.
– Wenn ich meine 32K-Ram in der P-Box habe, wirft mein TI mit Modulen, die dieses Ram benutzen, einige Speicherplätze nach ca. 40 Minuten durcheinander. Reinigen und Port auswechseln habe ich schon probiert. Ein sofortiger Neustart funktioniert nur, wenn ich die 32K-Ram entfernt habe und ohne diese arbeite – wer kann helfen?
– TI-Assembler ja, aber wie erlernt man das? Gibt es Bücher darüber?
Bernhard Betz
5000 Köln 21

Ihre Bemerkungen zu dem Siemens-Laufwerk werden sicher einigen unserer Leser helfen. (Siehe hier auch die Beantwortung eines anderen Briefes.)
Bei der 32K-Speichererweiterung wird es schon schwieriger. Da häufig Anfragen nach den Reparaturmöglichkeiten der Original-Ti-Teile kommen, hier einmal grundsätzlich die Bemerkung, daß diese Teile nur von Texas Instruments repariert werden. Wenn Sie zur Selbsthilfe greifen wollen, was nur erfahrene Elektronik-Bastler tun sollten, so sollten Sie erst einmal genau eruieren, welche Speicherstellen wie durcheinander geworfen werden. Dies kann man mit dem Debugger des Editor-Assembler machen. Vorausgesetzt, er läßt sich noch betreiben, da er ja als Maschinenprogramm in eben diese Ramerweiterung geladen wird. Da Ihr Fehler nur nach einer gewissen Zeit auftritt, kann es auch sein, daß sich ein Bauteil übermäßig erwärmt, dieses wäre dann defekt. Da zur Adressendekodierung in allen TI-Karten für die P-Box besonders programmierte Bausteine eingesetzt werden (sog. PAL's) ist eine Reparatur, wenn dieser PAL defekt ist, nur bei TI möglich. Assembler können Sie mit der TI-REVUE lernen, den ersten Bericht lesen Sie in diesem Heft.

UNSER ☎ TELEFONSERVICE ☎

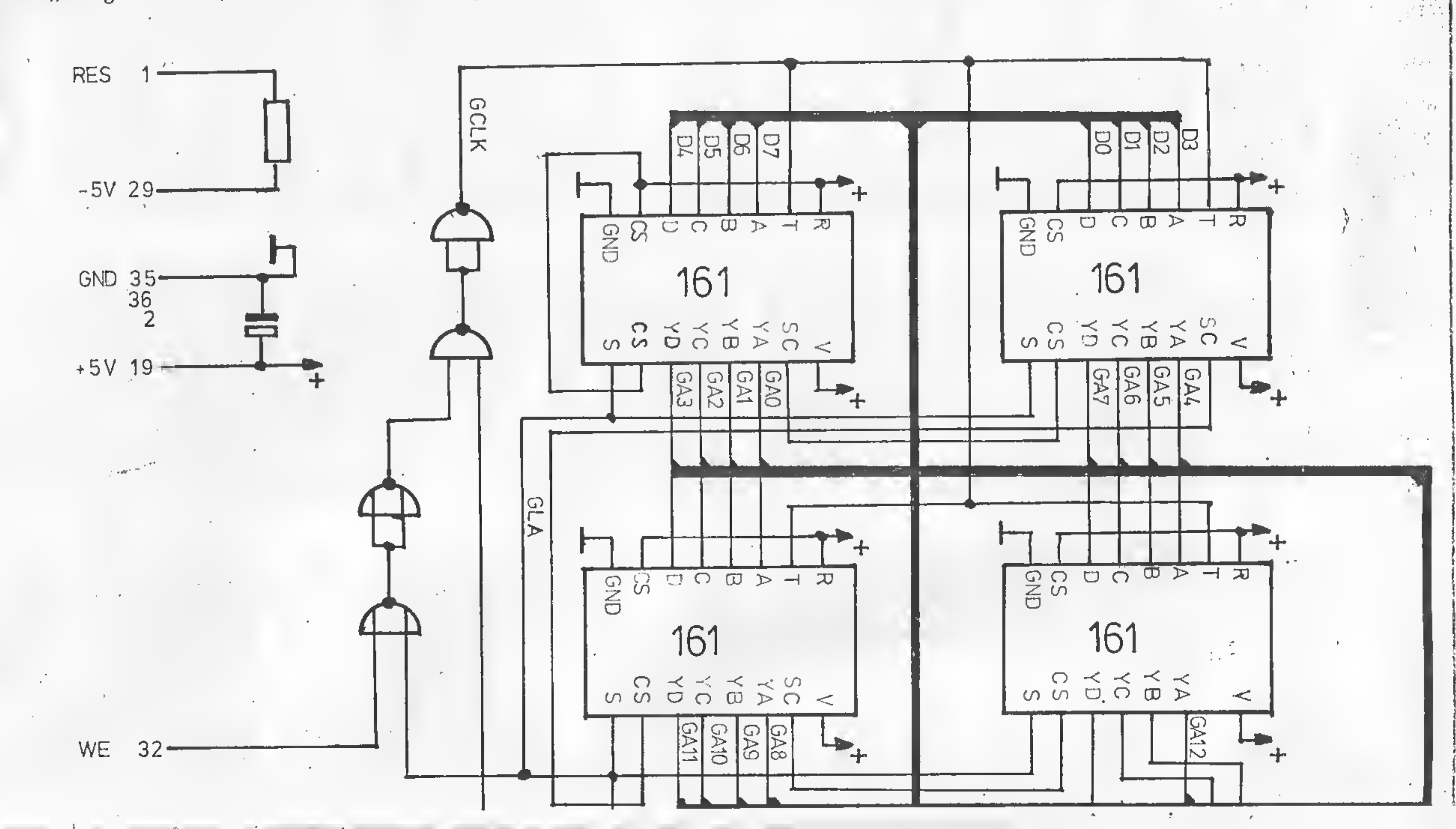
Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf dieser Seite

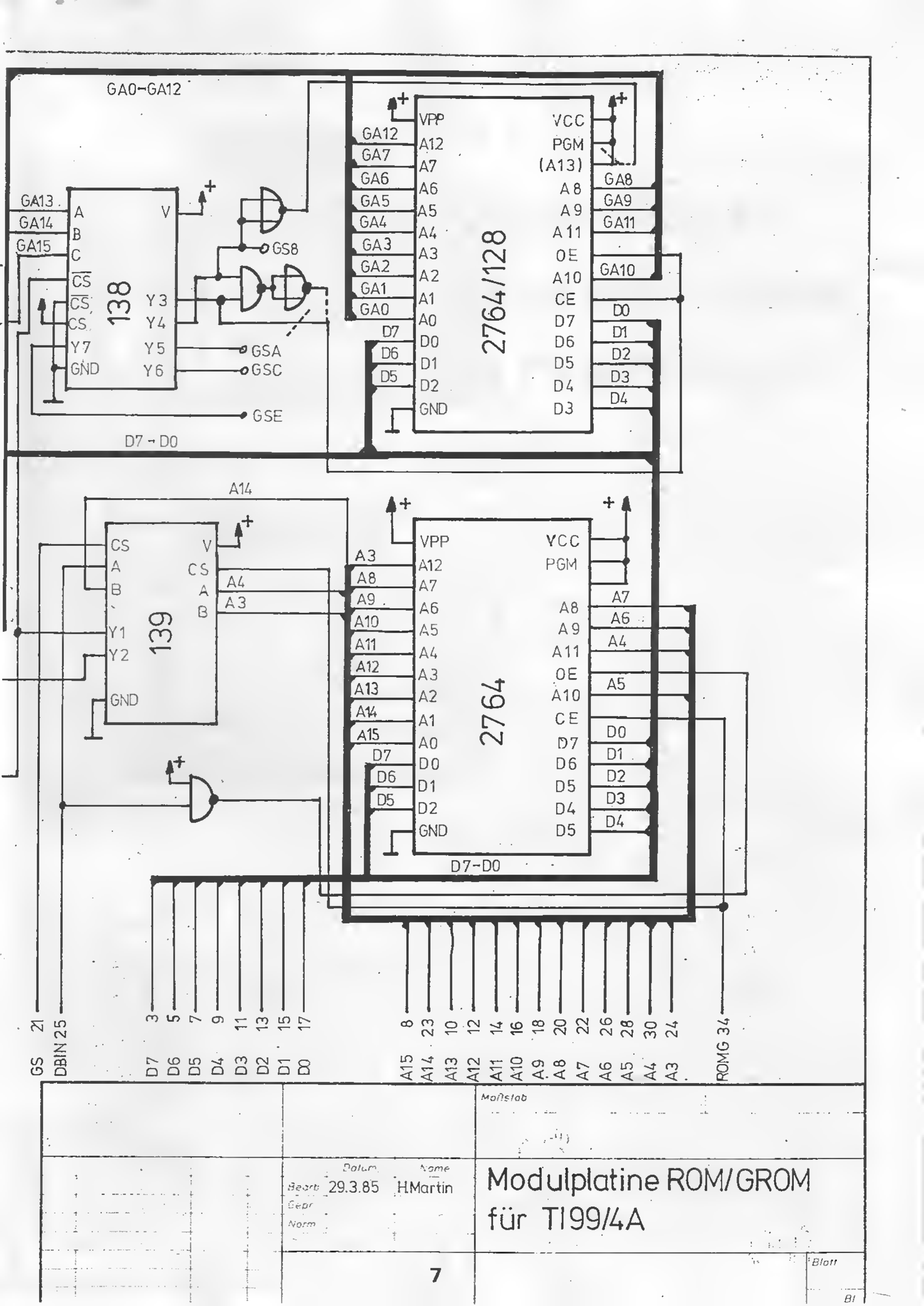
ACHTUNG!!!

Ab sofort steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33220 und für Listings/Programme: 089/286036.

MODULPLATINE FÜR DEN TI 99/4A

Schon seit längerer Zeit beschäftigen sich einige Leute mit einem der größeren Probleme des TI 99/4A für den Hardware-Bastler: Den GROM's. Zum Zeitpunkt der Produktionseinstellung begannen langsam die ersten nachgebildeten GROM's zu laufen. Mit dem Ausverkauf bei Texas Instruments kamen aber auch Module aus den „heiligen Hallen", die ein etwas merkwürdiges Innenleben aufwiesen: Ein eindeutiges GPL-Programm, das ist ja die Programmiersprache, die in den GROM's abgelegt ist, war da aus einem Eprom mit diversen TTL-IC's herauszulesen. Nun, die Schaltung war schnell abgezeichnet, aber es folgte nur ungläubiges Staunen: Da fehlte doch etwas; aber es funktionierte ja, zweifellos!



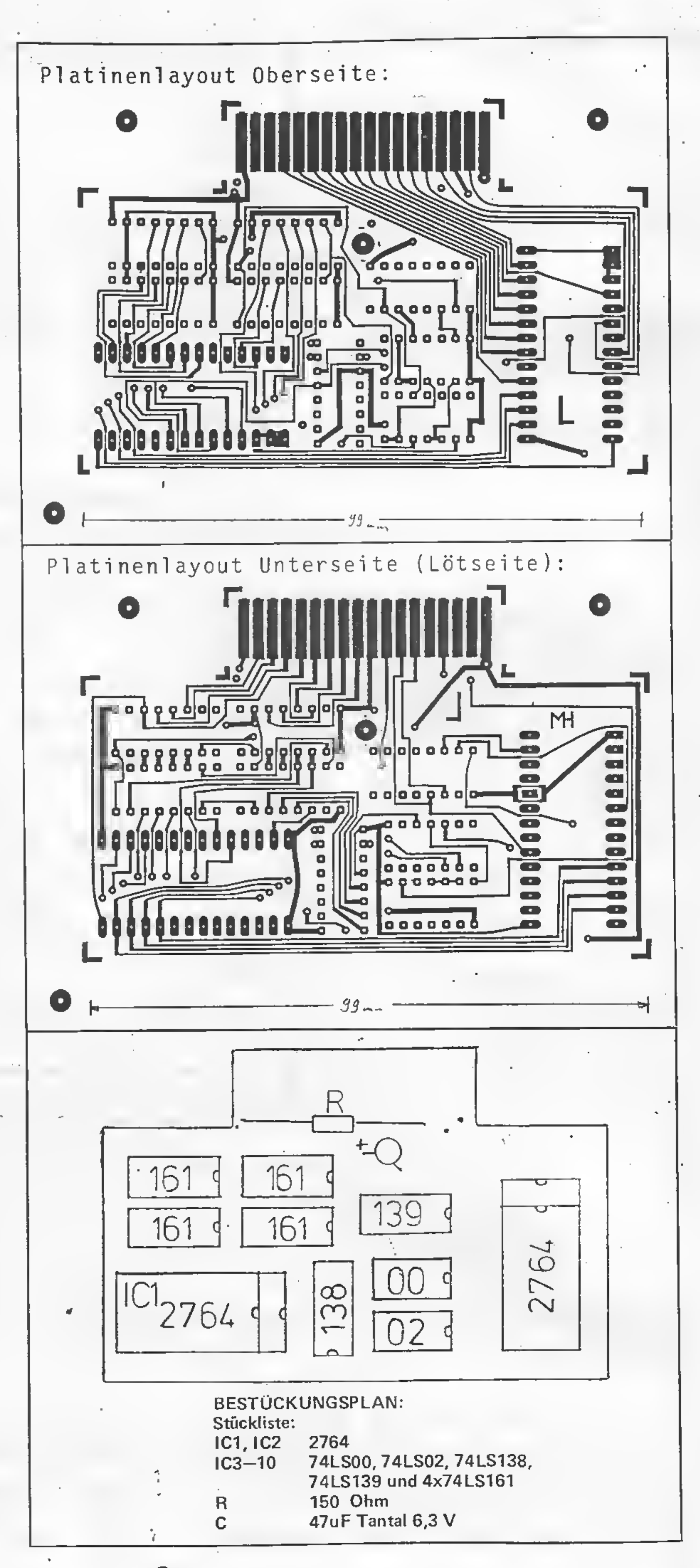


GA0–GA12
GA13
GA14
GA15
A
B
C
CS
CS
CS
138
V
Y3
Y4
Y5
Y6
Y7
GND
GS8
GSA
GSC
GSE
D7 – D0
2764/128
VPP
A12
A7
A6
A5
A4
A3
A2
A1
A0
D0
D1
D2
GND
VCC
PGM
(A13)
A8
A9
A11
OE
A10
CE
D7
D6
D5
D4
D3
GA12
GA7
GA6
GA5
GA4
GA3
GA2
GA1
GA0
GA8
GA9
GA11
GA10
A14
139
CS
Y1
Y2
A4
A3
2764
D7–D0
GS 21
DBIN 25
D7 3
D6 5
D5 7
D4 9
D3 11
D2 13
D1 15
D0 17
A15 8
A14 23
A13 10
A12 12
A11 14
A10 16
A9 18
A8 20
A7 22
A6 26
A5 28
A4 30
A3 24
ROMG 34
Maßstab
Datum
Name
Bearb. 29.3.85 H.Martin
Gepr.
Norm
Modulplatine ROM/GROM
für TI99/4A
Blatt
Bl
7

Was da fehlt? Jedes Grom besitzt zur Adressierung der gewünschten Speicherstelle einen internen Zähler, der über die Datenleitungen gesetzt werden kann, der automatisch bei jedem Lesen von Daten um 1 weiterzählt,und dessen Stand auch wieder ausgelesen werden kann. Und eben dieses Auslesen der Adresse fehlte. Ja, da dies funktioniert, bleibt nur als Erklärung, daß aus den intern im TI 99/4A enthaltenen drei GROM's die Adresse (auch eine, bei der sie normalerweise nicht ansprechen) ausgeben.
Dieses Fehlen des Lesens der Adresse machte die Schaltung viel einfacher und so konnte nun eine Platine für den Modulport entwickelt werden, die den dort zur Verfügung stehenden GROM- und ROM-Bereich abdeckt. Dabei wurde die Größe so gewählt, daß sie in die üblichen Modulgehäuse paßt.

DIE SCHALTUNG:

Der ROM-Bereich am Modulport reicht von >6000 bis >8000, umfaßt also insgesamt 8KByte, in dem Assembler-Programme abgelegt sind. Hierfür wurde auf der Platine ein 2764 Eprom vorgesehen, wobei auch ohne Änderung bei kleinen Programmen ein 2732 eingesetzt werden kann. Zusätzlich sind mit der zweiten Hälfte des 74LS139 die Adressen noch auf 2KByte Blöcke dekodiert, für Eigenkonstruktionen.
Der GROM-Bereich umfaßt die Grom-Adresse >6000 bis >FFFF. Zuerst ist für die Groms aber erst einmal ein Zähler aus 4 Stück 74LS161 vorhanden. Diese sind in der Zählerfunktion als ganz normaler 16Bit-Binärzähler geschaltet. Wird die Leitung GLA Low, d.h. aktiv (über die zweite Hälfte des 74LS139 aus den Signalen GS, DBIN und A14 dekodiert), übernehmen die oberen 8 Bits des Zählers den Stand der unteren 8 Bits und die unteren 8 Bits werden mit dem Wert auf den Datenleitungen geladen. Die Leitung GCLK wird bei jedem Lesezugriff aktiv und incrementiert den Adressenzähler. Die Ausgänge des Adressenzählers dienen als Adressenleitungen für ein bis 5 Eproms 2764. Dabei wird die Dekodierung in 8KByte-Blöcke von einem 74LS138 vorgenommen. Auf der Platine verdrahtet ist aber



BESTÜCKUNGSPLAN:
Stückliste:

IC1, IC2	2764
IC3–10	74LS00, 74LS02, 74LS138, 74LS139 und 4x74LS161
R	150 Ohm
C	47uF Tantal 6,3 V

nur das GROM mit der Adresse >6000 bzw. mit geringfügigen Änderungen ist auch ein 27128 für GROM >6000 und GROM> 8000 einzusetzen.

DIE BESTÜCKUNG:

Die Platine muß doppelseitig, am besten mit durchverkupferten Bohrungen, hergestellt werden. Die Seite mit dem Schriftzug HM ist dabei die untere Lötseite. Die Bestückung geht aus dem Plan hervor, so daß hier keine weiteren Ausführungen notwendig sind. Soll im GROM ein 27128 eingesetzt werden, so sind die im Schaltbild gestrichelt eingezeichneten Leiterbahnen zu verbinden bzw. aufzutrennen, d.h. die vom 74LS138 am Pin 12 abgehende Leitung und die Verbindung zwischen Pin 26 und 27 wird aufgetrennt. Die direkt an Pin 26 des Eprom führende Leitung wird mit etwas Lötzinn mit dem Pin 26 verbunden und die Bohrung neben Pin 15 des 74LS138 wird mit der Leitung zu den Pins 20 und 22 des Eproms verbunden. Sollen noch weitere Eproms auf den Grom-Adressen >A000, >C000 und >E000 eingebaut werden, so sind sie auf dem 27128 zu stapeln, d.h. das Eprom wird einfach aufgesetzt und alle Pins werden verlötet, außer Pin 20 und 22. Diese werden an jedem einzelnen Eprom verbunden und dann zu dem entsprechenden Ausgang des 74LS138 gelegt, also den im Schaltbild mit GSA, GSC und GSE bezeichneten Punkten.

DIE ANWENDUNG:

In den ROM-Bereich können eigenständige Assemblerprogramme gelegt werden. Dabei ist natürlich zu beachten, daß keinerlei der sonst vorhandenen Hilfsroutinen wie KSCAN, VMBW, VSBW usw. vorhanden sind. Auch muß der komplette Bildschirmaufbau mit dem Laden der Charaktertafeln usw. selbst vorgenommen werden. Der Einsprung vom Titelbild ist bei nahezu allen in Deutschland (außer den Konsolen, die sich im Titelbild mit (c) 1983 melden) möglich. Dazu muß folgender Kopf gesetzt werden:

```
AORG> 6000
DATA> AA01 Kopf und Version
DATA >0100 Anzahl Programme
DATA >0000 (Power-up Routine)
DATA PRGLNK Programm-Link
DATA >0000 (DSR-Link
DATA >0000 (Unterprogramm-Link)
DATA >0000 (Interrupt-Link)
PRGLNK DATA PRGLK2 (0 wenn
   kein 2. Programm)
        DATA ENTRY (Ein-
        sprungadresse)
        BYTE 4 (Namenslänge)
        TEXT "TEST" (Name)
```

Die im Kopf in Klämmern angegebenen Zeiger zu weiteren Listen sind im ROM des Modulportes nicht verwendbar, aber im GROM und auch bei den DSR-Routinen (z.B. Disk-Controller) wird genau der gleiche Kopf benutzt. Womit wir beim GROM sind. In diesem Bereich können eigene GPL-Programme gelegt werden, was ja jetzt, da die Geheimnisse um diese Programmiersprache gelüftet sind, möglich geworden ist. Erfolgt der Einsprung auf das Programm über das simulierte GROM, so ist das Programm auf jeder Konsole zu betreiben.

Diese Platine ermöglicht nun wirklich neue Programmentwicklungen für den TI 99/4A, die auf der gleichen Linie wie die originalen TI-Module liegen, ohne daß man weiterhin auf besondere Bauteile des ehemaligen Herstellers angewiesen ist. Ein erstes, neues GPL-Programm für diese Platine ist derzeit auch schon in Arbeit, es wandelt LIST-Dateien im Format Variable 80 wieder zurück in lauffähige Programme. Man könnte ja auch ein Programm zur Umwandlung des Basiccode in TI-Basic schreiben. Ein weiterer Anwendungszweck dieser Platine sei der Fantasie jedes einzelnen überlassen. Eine gewisse Nachfrage vorausgesetzt, bin ich gerne bereit, durchverkupferte Platinen herstellen zu lassen, allerdings müßte der Vertrieb dann aus Gründen des Aufwandes über einen Fachhändler erfolgen.

Heiner Martin

ASSEMBLER LEICHT GEMACHT FÜR DEN TI 99/4 A

Heute und in den nächsten Folgen wollen wir ein bißchen näher auf das Thema Maschinensprache mit dem TI 99/4A eingehen.
Dabei sollen nicht nur das Editor/Assembler Modul, sondern auch die Module Extended Basic und Mini-Memory angesprochen werden.
Beginnen wollen wir ganz einfach, aber am Schluß jedes Beitrages sollen auch für die Könner einige spezielle Tricks verraten werden.

Was brauchen wir, um mit Assembler arbeiten zu können? Auf jeden Fall eines der eben schon angesprochenen Module, wobei hier das neue Extended Basic II plus zum Extended Basic zählt, auch wenn es eine besonders nützliche Routine zum Abspeichern von Assembler Programmen auf Kassette besitzt, ähnlich wie das Mini-Memory.

DAS MINI-MEMORY:

Mit diesem Modul kann man allein mit der Konsole zusammen schon in Maschinensprache (Assembler) programmieren. Eine eventuell vorhandene Speichererweiterung ist ebenfalls nutzbar. Ein Line-by-Line Assembler, also ein Programm, welches die Bildschirmeingaben in direkt vom 9900-Prozessor verwendbaren Code umwandelt, gehört zum Lieferumfang dieses Moduls. Zum Abspeichern der Assemblerroutinen genügt beim Mini-Memory ein Kassettenrekorder, ein Abspeichern auf Diskette ist nicht möglich, jedoch können mit dem Mini-Memory vom Editor/Assembler auf Diskette erstellte Maschinenprogramme verarbeitet werden.

DAS EXTENDED BASIC:

Bei diesem Modul ist unbedingt eine Speichererweiterung notwendig. Das Modul enthält auch kein Assembler-Programm, welches ebenfalls noch benötigt wird, allerdings wurden schon mehrere derartige Programme veröffentlicht. Zusammen mit einem Diskettenlaufwerk können vom Editor/Assembler erstellte Assemblerprogramme benutzt werden. Problematisch ist das Abspeichern der mit einem Extended Basic Assembler erstellten Routinen. Hier sind nochmals Hilfsprogramme notwendig.

DER EDITOR/ ASSEMBLER:

Er ist ohne Zweifel das komfortabelste Modul für die Assemblerprogrammierung, verfügt es doch über einen Editor, mit dem der Quellcode deutlich lesbar erstellt wird. Dieser Quellcode wird von einem Assembler, der auch einen kompletten Ausdruck auf einem Drucker erlaubt, in den Objektcode umgewandelt, welcher dann wieder zum Abarbeiten geladen wird. Allerdings erfordert dieses Modul auch die größte Ausrüstung: Speichererweiterung und Diskettenlaufwerk.

GRUNDSÄTZLICHES

Beim TI 99/4A wird Assembler nicht mit dem „Hammer programmiert", wie bei vielen anderen Computern dieser Klasse. Ursprünglich dachte Texas Instruments wohl nur an den Editor/ Assembler, und hier ist das erstellte Maschinenprogramm so richtig komfortabel: Eine relative Adressierung ist möglich, d.h. der Benutzer muß sich nicht um den belegten Speicherplatz kümmern, sondern das besorgt das Programm, welches den Objektcode einfach in einen freien Speicherbereich lädt und alle notwendigen Adressen dabei umrechnet. Der Aufruf des Programms (LINK) erfolgt nun auch nicht über irgendeine Systemadresse, die man sich permanent merken muß, sondern über einen Namen, der bis zu 6 Stellen lang sein darf und der in einer gesonderten Tabelle zusammen mit der dazugehörigen Einsprungadresse abgelegt ist. Diese Benutzerfreundlichkeit bedarf aber nun einiger Übersicht bei direkter Assemblierung, wie es mit dem Line-by-Line Assembler des Mini-Memorys und mit fast allen Extended-Basic Assemblern geschieht.

DIE LITERATUR

Unabdingbar notwendig zum Programmieren des TI 99/4A in Assembler ist das Handbuch zum Editor/Assembler. Dies ist in Englisch geschrieben, es gibt davon aber auch eine zwar nicht als exzellent zu bezeichnende, aber doch hilfreiche Übersetzung ins Deutsche für diejenigen, die des Englischen nicht sehr mächtig sind. Eine weitere, sehr gute Lektüre zum Thema 9900 Assembler gibt es leider nicht mehr: Das 16 Bit Microprozessor Kursbuch von Texas Instruments. Ebenfalls sehr wichtig ist natürlich die Anleitung zum jeweils verwendeten Assembler (bei Extended Basic und Mini-Memory.

DAS ERSTE PROGRAMM

Aller Anfang ist schwer, besonders bei der Programmierung in Maschinensprache. Zwei der wesentlichen Schwierigkeiten mit Assembler wollen wir gleich zu Anfang behandeln: Der Einsprung auf das Programm und das Darstellen eines Zeichens auf dem Bildschirm. Folgendes Programm soll als Beispiel dienen:

```
        DEF   START
        REF   VSBW
MYWS    BSS   32
START   LWPI  MYWS
        LI    0,>0190
        LI    1,>4100
        BLWP  §VSBW
ENDLOS  LIMI  2
        LIMI  0
        JMP   ENDLOS
        END
```

Dieses Programm für das Editor/Assembler Modul wird unter dem Namen START aufgerufen, das beinhaltet die erste Zeile mit DEF. REF bedeutet Reference und ist eine weitere bedienungsfreundliche Funktion: Beim Laden des Programms wird

hier die REF/DEF-Tabelle abgesucht, ob der Begriff enthalten ist und wenn ja, wird der entsprechende Wert eingesetzt.
Dann werden mit BSS 32 Bytes (16 Worte) für den Workspace (Arbeitsbereich) des TMS 9900 reserviert. Dieser Prozessor besitzt ja nicht die sonst viel verbreiteten Register in der CPU, sondern benutzt dafür einen Bereich im Ram, auf den der Workspacecounter zeigt. Ab dieser Adresse werden 16 Register (R0 bis R15) benutzt. Als 16-Bit Prozessor umfassen diese Register natürlich je 2 Bytes, deshalb der Wert 32 bzw. >0020.
Das Programm beginnt damit, daß der Workspacecounter geladen wird, und zwar mit der Anfangsadresse des eben reservierten Speicherbereiches. Nun laden wir die Register R0 mit der Adresse, in die wir in das VDP-Ram schreiben wollen und das höherwertige Byte von R1 mit dem Wert >41, was dem ASCII Wert des Buchstabens A entspricht. Mit BLWP wird nun ein Unterprogramm aufgerufen. Solche Unterprogramme stellen alle Module, mit denen in Assembler gearbeitet werden kann, zur Verfügung, VSBW ist dabei eine Kurzform für VDP Single Byte Write, was soviel bedeutet wie ein Byte in das VDP-Ram schreiben.
Danach wird mit LIMI 2 der Interrupt freigegeben und mit LIMI 0 wieder unterbunden. Dies dient dazu, daß wir mit der Quit-Taste, die ja von der Interrupt-Routine abgefragt wird, aus der Endlosschleife zum Titelbild kommen können. Die Endlosschleife ist mit dem unbedingten Sprungbefehl JMP (Jump) realisiert.
Kommen wir nun zu den verschiedenen Assembler. Mit dem Editor/Assembler erstellen wir mittels des Editors die Quellcode-Datei und assemblieren diese dann mit dem Assembler. Dabei sollten wir die Dateien schon vom Namen her deutlich kennzeichnen, z.B. mit DSK1TESTQ (für Quellcode) und DSK1TESTC (für Objektcode). Bei der Abfrage OPTIONS des Assemblers geben wir RC oder nur R ein. Wird das C gewählt, wird ein sogenannter compressed Code erstellt, dieser kann vom Extended-Basic-Modul nicht geladen werden. Danach wählen wir den Abschnitt LOAD AND RUN, tippen DSK1. TESTC ein und starten das Programm über den Namen START.
Nun haben wir aber einen einfachen Assembler in Extended-Basic geschrieben oder den des Mini-Memorys. Hier sind ein paar Einschränkungen zu machen: DEF, REF und BSS sind meist nicht vorhanden. Wir beginnen also an einer freien Speicherstelle, merken uns die Adresse (z.B. >7000 für das Mini-Memory Adresse (z.B. >2500 unter X-Basic oder >7000 für das Mini-Memory) und springen 32 Bytes höher, d.h. auf >2500 bzw. >7D20.
nun beginnen wir mit dem Programm:

```
LWPI   >2500 (>7D00)
LI     R0,>0190
LI     R1,>4100
BLWP   §>2020 (>6024)
LIMI   2
LIMI   0
JMP    >2530 (>7D30)
END
```

Die Werte in Klammern gelten für das Mini-Memory. Wir haben also für den Aufruf des Unterprogramms eine feste Adresse eingegeben und dies ebenfalls beim Wert für den Sprung getan. Hier mußten wir uns während des Programmierens merken, auf welche Adresse wir springen wollten. Weiter haben wir uns die Adresse des Programmbeginns (>2520 bzw. >7D20) und der ersten freien Adresse hinter dem Programm gemerkt (>253A bzw. >7D3A);
Jetzt können wir aber das Programm noch nicht aufrufen und so müssen wir die dafür notwendigen Zeiger noch entsprechend laden.

Zuerst müssen wir das Programm gegen Überschreiben schützen und den Zeiger auf den Beginn des freien Rams laden. Beim Extended Basic ist die Adresse >2002 mit >253A und beim Mini-Mem ist die Speicherzelle >701C mit >7D3A zu laden. Am einfachsten geschieht dies auch mit dem Line-by-Line Assembler mit AORG> 2002 (> 701C für Mini-Mem) und dann DATA >2536 (>7D36).
Nun müssen wir noch die DEFinitionstabelle und den Zeiger darauf laden. Unter Extended-Basic ist dies die Speicherstelle >2004 (> 701E für Mini-Mem). Von dem Wert, der bisher in dieser Speicherstelle steht, ziehen wir 8 ab und geben diesen Wert dann ein, also z.B.> 3FF8 bei Extended Basic oder >7FF8 bei Mini-Mem. Die letzte Stelle dieses Zeigers ist immer eine 8 oder eine 0.
Dann gehen wir an die Adresse, deren Wert wir gerade eingegeben haben mit z.B. AORG> 3FF8 (>7FF8) und nun wird es kompliziert. Jetzt müssen wir mittels DATA den Namen in ASCII eingeben. Bei dem Namen Start ergibt sich >5354, >4152, >5420. Bitte beachten, der Name ist immer 6 Stellen lang, notfalls mit >20 (entspricht dem Leerzeichen) füllen. Darauf folgt nun die Startadresse des Programms also >2520 bzw.> 7D20 beim Mini-Mem.
Jetzt können wir das Programm mit dem Namen aufrufen und ausführen.

In Extended Basic bzw. im Basic geschieht dies mittels CALL LINK ("START"). Jetzt funktioniert das Programm über LOAD AND RUN des Mini-Mems und des Assembler-Moduls einwandfrei, wir sehen das A deutlich auf dem Bildschirm, aber aus dem Basic oder Extended Basic aufgerufen, sehen wir gar nichts. Das liegt an einer Eigenheit, wie im Basic der Bildschirm aufgebaut ist: Hier sind alle Zeichen mit einem Offset von> 60 abgebildet. Soll das Programm also unter Basic arbeiten, muß die entsprechende Zeile heißen:

```
LI    R1,>A100
```

Das war es dann erst einmal für heute. Übrigens könnt Ihr mir auch gerne schreiben, wenn Ihr ganz besondere Probleme mit der Assembler-Programmierung habt, aber bitte nicht zur Fehlersuche in 20KByte langen Programmen! Auf die Anwendungs des Debuggers werden wir hier auch noch eingehen.

FÜR EXPERTEN

Bleibt noch unser Tip für die Assembler-Könner unter Euch: Im Betriebssystem der RS232-Karte steckt eine Interrupt-Routine, die ein interruptgesteuertes Empfangen von Zeichen erlaubt. Dazu müssen folgende Pointer gesetzt werden: Auf> 8300 befindet sich der Pointer zum PAB-Buffer im VDP, auf> 8302 (Byte) die maximale Datensatzlänge, auf> 8303 (Byte) die Länge des zu empfangenden Datensatzes und auf> 8304 (Byte) die aktuelle Länge des Datensatzes. Im Falle eines Empfangserrors ist das letzte empfangene Zeichen> FF. Hat der Datensatz die auf >8303 angegebene Länge erreicht, wird als empfangenes Zeichen> FE in den PAB Buffer geschrieben. Um das Ganze zu aktivieren, wird ganz normal eine Datei eröffnet, nur mit dem Unterschied, daß in der PAB beim OP-Code das I. Bit (MSB) gesetzt werden muß, also z.B. 80 für OPEN. Benötigt wird das zum Beispiel für Terminal Emulator Programme. *Heiner Martin*

PHYSIK

Dieses Programm beinhaltet eine umfangreiche Formelsammlung aus vielen Bereichen der Physik. Verschiedenartige physikalische Probleme werden miteinander verknüpft, so daß sich komplette Aufgaben lösen lassen. Die nach abgeschlossener Berechnung erscheinenden Merksätze dienen vorwiegend der Unterstützung, um auch bei anderen Aufgaben den richtigen Ansatz zu finden.
Die zur Wahl stehenden Themen sind unter anderem:
Geschwindigkeit, Beschleunigung, geradlinige Bewegung, freier Fall, waagrechter Wurf, schräger Wurf, Kraft und Masse, schiefe Ebene, Reibung, Arbeit, Energie und Leistung, Dreh- und Kreiselbewegung, Fliehkraft, Gravitation, Fallbeschleunigung, Satellitengeschwindigkeit, Faden- und Federpendel.
Wird das Programm gestartet, erscheint zunächst ein Animationsbild. Durch Drücken einer beliebigen Taste folgt das Titelbild. Ein weiterer Tastendruck führt zum Hauptmenü. Von nun an kommen Sie durch Eingabe der Kennziffer und "ENTER" in die einzelnen Unterprogramme und auf dem gleichen Weg von dort aus zu den gewünschten Berechnungen. *Rudolf Trumbach*

Programm:	*Physik*
Autor:	*Rudolf Trumbach, Mauritiusstraße 4a, 4630 Bochum 1*
Kategorie:	*Lernprogramm & Anwenderprogramm*
Sprache:	*Ext.-Basic*
Ben. Zubehör:	*Konsole, XBasic-Modul*
Speicherbelegung:	*12965 Bytes*
Optimierung:	*Gut*

Kurzbeschreibung: *Dieses Programm beinhaltet eine überaus reichhaltige Formelsammlung aus vielen Anwendungsbereichen der Physik. Eine sehr gute Benutzerführung macht es auch dem Laien möglich, mit diesem Programm zu arbeiten.*
Bemerkung: *Der Autor hat, laut eigenen Angaben, 3 Monate an diesem Programm gearbeitet. Diese Zeit war nicht vergeudet, ein professionelles Anwenderprogramm ist entstanden. Der äußere Rahmen und der sachliche Inhalt lassen keinerlei Kritik zu. Die vielen Formeln brachten uns beim Austesten ganz schön ins Schwitzen.*

BCOPY

Dieses Unterprogramm können Sie an jedes Hauptprogramm mit der MERGE-Option anbinden.
Im Hauptprogramm wird es mit 'CALL BCOPY' aufgerufen und erstellt Ihnen eine Kopie Ihres Bildschirmes, vorausgesetzt, Sie verfügen über einen Seikosha 100A. Nachdem das Hauptprogramm an das Unterprogramm abgegeben hat, ertönt ein kurzes Signal. Drücken Sie nun die Taste 'N', so bekommen Sie einen Ausdruck in normaler Breite. Die Taste 'J' weist den Drucker an, den Ausdruck in doppelter Breite anzufertigen. Da der Bildschirm nur eine Spaltenbreite von 32 Zeichen und der Drucker eine Breite von 80 Zeichen darstellen kann, wird Ihnen nun durch ein zweites Signal die Möglichkeit geboten, die restlichen, freibleibenden Spalten mit einem Zeichen Ihrer Wahl zu füllen. Drücken Sie nach dem 2. Signal eine Taste, so wird

Fortsetzung auf Seite 18

```
100 ! *********************
110 ! *                   *
120 ! *      PHYSIK       *
130 ! *   Copyright by    *
140 ! * Rudolf  Trumbach  *
150 ! *                   *
160 ! *   Bearbeitet vom  *
170 ! *     Team des      *
180 ! * Aktuell Verlages  *
190 ! *     Muenchen      *
200 ! *                   *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole *
230 ! *    Ext. Basic     *
240 ! *                   *
270 ! *  Speicherbelegung *
280 ! *    12965 Bytes    *
290 ! *********************
295 !
300 ON WARNING NEXT :: ON BR
EAK NEXT
310 CALL CLEAR :: RANDOMIZE
:: CALL AFARBE(10,2):: CALL
HCHAR(24,28,62,2)
320 DATA 80,72,89,83,73,75,8
0,72,89,83,73,75,80,72,89,83
,73,75,80,72,89,83,73,75,80
330 RESTORE 320 :: FOR Z=2 T
O 26 :: READ A :: CALL MAGNI
FY(2)
340 CALL SPRITE(#Z,A,INT(Z/3
)+3,92,114,Z*INT(RND*4.5)-2.
25+Z/2*SGN(RND-.5),Z*INT(RND
*4.5)-2.25+Z/2*SGN(RND-.5)):
: CALL SOUND(-50,1500,0)
350 IF Z=INT(RND*27)+Z THEN
CALL MAGNIFY(1):: CALL SOUND
(-100,1400,0,1900,0,1600,0)
360 CALL KEY(0,K,ST):: IF ST
<>0 THEN 410
370 NEXT Z :: GOTO 330
380 !
390 !TITELBILD+CHAR
400 !
410 CALL CLEAR :: CALL DELSP
RITE(#1):: CALL BFARBE(1,2):
: FOR Z=1 TO 3 :: CALL HCHAR
(Z,3,42,28):: NEXT Z
420 CALL MAGNIFY(4):: CALL S
OUND(-5,110,0,120,0,115,0)
430 FOR Z=22 TO 24 :: CALL H
CHAR(Z,3,42,28):: NEXT Z ::
CALL VCHAR(4,3,42,42):: CALL
 VCHAR(4,29,42,42)
440 CALL CHAR(96,RPT$("F",16
),97,"0103070F1F3F7FFF",98,"
FFFEFCF8F0E0C080",99,"FF7F3F
1F0F070301")
```

```
450 CALL CHAR(100,"80C0E0F0F
8FCFEFF",127,"000000FF",128,
"FF818181818181FF",37,"E020E
080E")
460 RESTORE 470 :: FOR Z=1 T
O 29 :: READ AA,BB,CC,DD ::
CALL VCHAR(AA,BB,CC,DD):: NE
XT Z
470 DATA 7,6,96,5,7,7,96,1,9
,7,96,1,7,8,96,3,7,10,96,5,9
,11,96,1,7,12,96,5,7,14,96,3
480 DATA 9,15,96,3,7,16,96,3
,7,19,96,1,7,20,96,1,7,18,96
,3,9,19,96,1,9,20,96,1
490 DATA 10,20,96,2,11,18,96
,1,11,19,96,1,7,22,96,5,7,24
,96,5,7,26,97,1,7,27,98,1
500 DATA 8,25,97,1,8,26,98,1
,9,25,96,1,10,25,99,1,10,26,
100,1,11,26,99,1,11,27,100,1
510 DISPLAY AT(16,6)SIZE(18)
:"ßMITßDEMßTIß99ö4Aß"
520 DISPLAY AT(2,5)SIZE(20):
"ßTEXASßßINSTRUMENTSß"
530 DISPLAY AT(23,4)SIZE(22)
:"ß1984ßRUDOLFßTRUMBACHß" ::
 CALL DELSPRITE(ALL):: CALL
CFARBE(13)
540 FOR Z=0 TO 30 STEP 3 ::
CALL SOUND(-100,1000,Z,1010,
Z,990,Z):: NEXT Z
550 IF PE=9 THEN 570
560 CALL KEY(0,K,S):: IF S<>
0 THEN 610 :: CALL CFARBE(13
):: CALL SOUND(1000,110,30):
: CALL CFARBE(2):: GOTO 560
570 DISPLAY AT(20,6)SIZE(18)
:"ßAUFßßWIEDERSEHENß" :: FOR
 I=1 TO 8
580 DISPLAY AT(18,6)SIZE(18)
:" " :: FOR Z=1 TO 50 :: NEX
T Z
590 DISPLAY AT(18,6)SIZE(18)
:"ßßßPROGRAMMENDEßßß" :: CAL
L SOUND(-100,2000,0):: FOR Z
=1 TO 50 :: NEXT Z
600 NEXT I :: FOR Z=1 TO 100
0 :: NEXT Z :: CALL CLEAR ::
 END
610 M=0 :: G=9.81 :: GK=6.67
E-11
620 !
630 ! HAUPTMENUE
640 !SP+
650 CALL CLEAR :: CALL AFARB
E(16,2):: CALL MAGNIFY(2)
660 RESTORE 670 :: CALL WRIT
E(1,15,3)
670 DATA HAUPTMENU,,1> MECH
ANIK,,2> SCHWINGUNGLEHRE,,3>
 PROGRAMMENDE
680 GOSUB 2250 :: ON K GOTO
730,1950,690,680,680,680,680
690 PE=9 :: GOTO 410
700 !
710 !WAHL MECHANIKPROGRAMME
720 !
730 RESTORE 740 :: CALL WRIT
E(1,15,3)
740 DATA MECHANIK,,1> BAHNBE
WEGUNG,2> KRAFT U. MASSE,3>
ARBEIT U. ENERGIE,4> DREH- U
. KREISBEWEGUNG,5> GRAVITATI
ON,6> HAUPTMENUE
750 GOSUB 2250
760 ON K GOTO 800,840,870,90
0,930,660,750
770 !
780 !MECHANIK 1-5
790 !
800 RESTORE 810 :: CALL WRIT
E(1,15,3):: GOSUB 2250
810 DATA BAHNBEWEGUNG,,1> GE
SCHWINDIGKEIT,2> GERADLINIGE
 BEWEGUNG,3> FREIER FALL
820 DATA 4> WAAGR. WURF,5> S
CHRAEGER WURF,6> RUECKSPRUNG
830 CALL DELSPRITE(ALL):: ON
 K GOTO 1010,1140,1270,1350,
1400,730,800
840 RESTORE 850 :: CALL WRIT
E(1,15,3)
850 DATA KRAFT U. MASSE,,1>
GEWICHTSKRAFT,2> HANGABTRIEB
SKRAFT,3> NORMALKRAFT,4> REI
BUNGSZAHL,5> REIBUNGSKRAFT,6
> RUECKSPRUNG
860 GOSUB 2250 :: CALL DELSP
RITE(ALL):: ON K GOTO 1490,1
500,1510,1540,1530,730,860
870 RESTORE 880 :: CALL WRIT
E(1,15,3)
880 DATA ARBEIT U. ENERGIE,,
1> ENERGIEERHALTUNGSSATZ,"..
.ARBEIT,LAGEENERGIE","...UND
 BEWEGUNGSENERGIE","...(NUR
HINWEISE)",,2> RUECKSPRUNG
890 GOSUB 2250 :: CALL DELSP
RITE(ALL):: ON K GOTO 1630,7
30,890,890,890,890,890
900 RESTORE 910 :: CALL WRIT
E(1,15,3)
```

```
910 DATA DREH- U. KREISBEWEG
UNG,,1> FREQUENZ,2> UMLAUFDA
UER,3> WINKELGESCHWINDIGKEIT
,4> BAHNGESCHW.,5> FLIEHKRAF
T,6> RUECKSPRUNG
920 GOSUB 2250 :: CALL DELSP
RITE(ALL):: ON K GOTO 1710,1
720,1730,1740,1750,730,920
930 RESTORE 940 :: CALL AFAR
BE(16,2):: CALL WRITE(1,15,3
)
940 DATA GRAVITATION,,1> GRA
VITATIONSKRAFT,2> FALLBESCHL
EUNIGUNG,3> 'LOOPINGPROBLEM'
,4> SATELLITENGESCHW.,"...FU
ER BAHN U. WINKEL"
950 DATA 5> RUECKSPRUNG
960 GOSUB 2250
970 CALL DELSPRITE(ALL):: ON
 K GOTO 1790,1820,1840,1880,
730,960,960
980 !
990 !RECHNEN MECHANIK 1-5
1000 !
1010 DISPLAY AT(13,3)ERASE A
LL:"GESCHWINDIGK.=" :: CALL
HCHAR(13,20,127,5)
1020 DISPLAY AT(12,19):"WEG"
 :: DISPLAY AT(14,19):"ZEIT"
 :: DISPLAY AT(22,1):"<1> GE
SCHWINDIGKEIT":"<2> WEG":"<3
> ZEIT" :: GOSUB 2370
1030 GOSUB 2280 :: ON K GOTO
 1040,1070,1090,1030,1030,10
30,1030
1040 GOSUB 2300 :: GOSUB 232
0
1050 V=S/T :: GOSUB 2420
1060 GOSUB 2340 :: IF K$="J"
 THEN 1010 ELSE 800
1070 GOSUB 2330 :: GOSUB 232
0
1080 S=V*T :: GOSUB 2430 ::
GOTO 1060
1090 GOSUB 2300 :: GOSUB 233
0
1100 T=S/V :: GOSUB 2440 ::
GOTO 1060
1110 GOSUB 2320 :: GOSUB 233
0 :: A=V/T
1120 DISPLAY AT(10,1)ERASE A
LL:"BESCHLEUNIGUNGSMOMENTANW
ERT":"":A;"M/S%":"":"BEI GLE
ICHMAESSIG BESCHLEU-":"":"NI
GTER BEWEGUNG AUS D. STAND"
1130 GOTO 1190
```

```
1140 RESTORE 1150 :: CALL WR
ITE(1,15,1):: GOSUB 2370 ::
GOSUB 2280
1150 DATA 1> BESCHLEUNIGUNG,
2> BESCHLEUNIGUNGSSTRECKE,3>
 GESCHW.NACH ABLAUF D.ZEIT,4
> GESCHW. NACH ABSOLVIERTER
  BESCHL.-STRECKE,
1160 DATA 5> WIE 4. MIT ANFA
NGSGESCW.,6> BREMSZEIT AUF 0
 KM/H,7> BREMSWEG AUF 0 KM/H
1170 ON K GOTO 1110,1180,120
0,1220,1240,1250,1260
1180 GOSUB 2320 :: GOSUB 241
0 :: S=(A/2)*T*T :: GOSUB 24
30 :: DISPLAY AT(13,1):"WENN
 AUS D. STAND":"":"BESCHLEUN
IGT WIRD"
1190 GOSUB 2340 :: IF K$<>"J
" THEN 800 ELSE 1140
1200 GOSUB 2410 :: GOSUB 232
0 :: V=A*T :: GOSUB 2420
1210 DISPLAY AT(15,1):"BESIT
ZT DAS FAHRZEUG EINE":"":"AN
FANGSGESCHWINDIGKEIT, IST":"
":"DIESE NOCH ZU ADDIEREN" :
: GOTO 1190
1220 GOSUB 2300 :: GOSUB 241
0 :: V=SQR(2*A*S):: GOSUB 24
20
1230 DISPLAY AT(15,1):"NACH"
;S;"METER":"":"GLEICHMAESSIG
ER":"":"BESCHLEUN. AUS DEM S
TAND" :: GOTO 1190
1240 GOSUB 2330 :: GOSUB 241
0 :: GOSUB 2300 :: V=SQR(V*V
+2*A*S):: GOSUB 2420 :: GOTO
 1190
1250 GOSUB 2330 :: GOSUB 241
0 :: T=V/A :: GOSUB 2440 ::
GOTO 1190
1260 GOSUB 2330 :: GOSUB 241
0 :: S=V*V/(2*A):: GOSUB 243
0 :: GOTO 1190
1270 RESTORE 1280 :: CALL WR
ITE(1,15,2):: GOSUB 2370
1280 DATA 1> FALLHOEHE,,2> G
ESCHWINDIGKEIT NACH,"...DURC
HFALLEN EINER","...BESTIMMTE
N STRECKE",,3> GESCHW. NACH
ABLAUF
1290 DATA "...EINER BESTIMMT
EN ZEIT"
1300 GOSUB 2280 :: ON K GOTO
 1310,1340,1330,1300,1300,13
00,1300
```

```
1310 GOSUB 2320 :: S=(G/2)*T
*T :: GOSUB 2430
1320 GOSUB 2340 :: IF K$="J"
 THEN 1270 ELSE 800
1330 GOSUB 2320 :: V=G*T ::
GOSUB 2420 :: GOTO 1320
1340 GOSUB 2300 :: V=SQR(2*G
*S):: GOSUB 2420 :: GOTO 132
0
1350 GOSUB 2300 :: GOSUB 233
0 :: S=V*SQR((2*S)/G):: GOSU
B 2430 :: GOSUB 2380
1360 CALL BFARBE(2,11):: RES
TORE 1370 :: CALL WRITE(10,2
2,1):: GOSUB 2450
1370 DATA DURCHFAELLT EIN KO
ERPER,SENKRECHT EINE STRECKE
,ODER WIRD ER VOM GLEICHEN,P
UNKT AUS WAAGRECHT,"ABGEWORF
EN,SO IST DIE"
1380 DATA ZEIT BIS ZUM AUFPR
ALL,FUER BEIDES GLEICH !
1390 GOSUB 2500 :: GOSUB 238
0 :: CALL AFARBE(16,2):: GOT
O 800
1400 GOSUB 2330 :: GOSUB 246
0
1410 DISPLAY AT(10,1)ERASE A
LL:"1> WURFZEIT":"":"2> WURF
WEITE":"":"3> STEIGHOEHE" ::
 GOSUB 2370
1420 GOSUB 2280 :: ON K GOTO
 1430,1470,1460,1420,1420,14
20,1420
1430 T=2*V*SIN(A)/G :: GOSUB
 2440 :: GOSUB 1480
1440 DISPLAY AT(17,1):"DIE Z
EIT BIS ZUM HOCHPUNKT":"":"D
ER WURFBAHN BETRAEGT GENAU":
"":"DIE HAELFTE"
1450 GOSUB 2340 :: IF K$="J"
 THEN 1400 ELSE 800
1460 S=V*V*(SIN(A)^2)/(2*G):
: GOSUB 2430 :: GOTO 1450
1470 S=2*V*V*COS(A)*SIN(A)/G
 :: GOSUB 2430 :: GOSUB 1480
 :: GOTO 1450
1480 DISPLAY AT(2,1):"VORAUS
SETZUNG:":"":"ABWURF- UND AU
FTREFFPUNKT":"":"LIEGEN IN E
INER WAAGR. EBENE":"":"DANN
IST DAS ERGEBNIS" :: RETURN
1490 GOSUB 2400 :: F=M*G ::
GOTO 1520
1500 GOSUB 2400 :: GOSUB 246
0 :: F=M*G*SIN(A):: GOTO 1520
1510 GOSUB 2400 :: GOSUB 246
0 :: F=M*G*COS(A)
1520 DISPLAY AT(10,1)ERASE A
LL:"KRAFT=";F;"NEWTON" :: GO
SUB 2380 :: GOTO 840
1530 GOSUB 2400 :: GOSUB 246
0 :: DISPLAY ERASE ALL:"WIE
LAUTET DIE REIBUNGSZAHL":""
:: INPUT "- WERT:":MU :: F=M
*G*COS(A)*MU :: GOTO 1520
1540 RESTORE 1570 :: CALL WR
ITE(10,20,1)
1550 DISPLAY AT(22,15):"H":"
REIBUNGSZAHL="&RPT$(CHR$(127
),3)&" ODER TAN(A)" :: DISPL
AY AT(24,15):"B"
1560 FOR Z=0 TO 6 :: CALL HC
HAR(1+Z,10-Z,96,1+Z):: CALL
HCHAR(1+Z,10-Z,97):: NEXT Z
:: CALL HCHAR(4,5,76):: CALL
 HCHAR(5,11,72):: Z=12
1570 DATA "BLEIBT EIN KOERPE
R NOCH GE-",RADE AUF DER SCH
IEFEN EBENE,"LIEGEN,KANN MAN
 SAGEN,DIE"
1580 DATA REIBUNGSKRAFT IST
GLEICH,DER HANGABTRIEBSKRAFT
.,DARAUS FOLGT :
1590 CALL HCHAR(8,8,66):: DI
SPLAY AT(1,Z):"EINGABE(METER
)" :: CALL HCHAR(2,14,127,15
):: DISPLAY AT(4,Z):"HOEHE:"
 :: DISPLAY AT(6,Z):"BASIS:"
1600 ACCEPT AT(4,18)VALIDATE
(NUMERIC)BEEP:H
1610 ACCEPT AT(6,18)VALIDATE
(NUMERIC)BEEP:B
1620 MU=H/B :: DISPLAY AT(23
,14)BEEP:MU :: CALL HCHAR(22
,17,127,11):: CALL HCHAR(24,
17,127,11):: GOSUB 2380 :: G
OTO 840
1630 CALL CLEAR :: CALL BFAR
BE(2,11):: RESTORE 1640 :: C
ALL WRITE(9,23,3):: GOSUB 25
00
1640 DATA DIE GROESSE DER EN
ERGIE,BLEIBT BEI ALLEN UM-,W
ANDLUNGEN ERHALTEN !,
1650 DATA "KRAFT.....= M*A"
,"ARBEIT.....= M*A*S*COS(A)"
,"LAGEENERGIE= M*G*H","BEWEG
UNGSE.= 1/2*M*V%"
1660 GOSUB 2450 :: GOSUB 238
0 :: RESTORE 1670 :: CALL WR
ITE(1,23,3):: GOSUB 2380
```

```
1670 DATA LIEGT Z.B. EIN KOE
RPER,"IN EINER GEWISSEN HOEH
E,",BESITZT ER EINE BESTIMMT
E,LAGEENERGIE. FAELLT ER,"HE
RUNTER, BLEIBT DIESE"
1680 DATA "ENERGIE, GENAUSO
WIE EINE",AN DIESEM KOERPER
VER-,"RICHTETE ARBEIT, ALS B
E-",WEGUNGSENERGIE ERHALTEN.
1690 DATA DURCH GLEICHSETZEN
 LAESST,SICH 'V' O. 'H' BEST
IMMEN.,(VERGL. AUCH FREIER F
ALL)
1700 CALL AFARBE(16,2):: GOT
O 870
1710 GOSUB 2470 :: GOSUB 232
0 :: N=GU/T :: DISPLAY ERASE
 ALL:"FREQUENZ:":"":N:"":"UM
DREHUNGEN PRO SEK." :: GOTO
1760
1720 GOSUB 2480 :: D=1/N ::
DISPLAY ERASE ALL:"PERIODEND
AUER:";D:"":"SEK. PRO UNDREH
UNG" :: GOTO 1760
1730 GOSUB 2480 :: O=2*PI*N
:: DISPLAY ERASE ALL:"WINKEL
GESCHW.:":"":O;"RAD/SEK." ::
 GOTO 1760
1740 GOSUB 2480 :: GOSUB 230
0 :: V=2*PI*N*S :: DISPLAY E
RASE ALL:"BAHNGESCHW.:":"":V
;"METER/SEK." :: GOTO 1760
1750 GOSUB 2480 :: GOSUB 230
0 :: GOSUB 2400 :: F=M*(2*PI
*N)^2*S :: DISPLAY ERASE ALL
:"FLIEHKRAFT:":"":F;"NEWTON"
1760 GOSUB 2380 :: GOTO 900
1770 F=M*V*V/S :: DISPLAY AT
(19,15):"=";F
1780 GOSUB 2340 :: IF K$="J"
 THEN 900 ELSE 730
1790 GOSUB 2400 :: DISPLAY E
RASE ALL:"DIE ZWEITE MASSE ?
":"" :: INPUT "- IN KG :":M2
 :: GOSUB 2300
1800 F=GK*M*M2/(S*S):: DISPL
AY AT(10,1)ERASE ALL:"DIE KR
AFT MIT DER SICH DIE":"":"MA
SSEN ANZIEHEN BETRAEGT":"":F
;"NEWTON"
1810 GOSUB 2340 :: IF K$="J"
 THEN 930 ELSE 730
1820 GOSUB 2400 :: GOSUB 230
0
1830 GF=GK*M/S^2 :: DISPLAY
AT(10,1)ERASE ALL:"DIE FALLB
ESCHL. DIESES":"":"HIMMELSKO
ERPERS BETRAEGT":"":GF;"M/S%
" :: GOTO 1810
1840 GOSUB 2300 :: V=SQR(G*S
):: GOSUB 2420 :: DISPLAY AT
(15,1):"MINDESTGESCHWINDIGKE
IT !" :: GOSUB 2380 :: CALL
BFARBE(2,11)
1850 RESTORE 1870 :: CALL WR
ITE(10,20,2):: GOSUB 2450 ::
 GOSUB 2500 :: GOSUB 2380
1860 CALL AFARBE(16,2):: GOT
O 930
1870 DATA DAMIT EIN FAHRZEUG
 NICHT,"AUS DEM LOOPING FAEL
LT,",MUSS DIE FLIEHKRAFT GLE
ICH,DER GEWICHTSKRAFT SEIN.,
,MINDESTVORAUSSETZUNG !!
1880 GOSUB 2400 :: GOSUB 230
0 :: V=SQR(GK*M/S):: O=SQR(G
K*M/S^3):: GOSUB 2420 :: DIS
PLAY AT(15,1):"DIE WINKELGES
CHW.":"":O;"RAD/SEK."
1890 GOSUB 2380 :: CALL BFAR
BE(2,11):: RESTORE 1900 :: C
ALL WRITE(10,16,3):: GOSUB 2
500 :: GOSUB 2450
1900 DATA BESCHREIBT EIN KOE
RPER,"EINE KREISBEWEGUNG, SO
",IST DIE ANZIEHUNGSKRAFT,GL
EICH DER FLIEHKRAFT !
1910 GOSUB 2380 :: GOTO 930
1920 !
1930 !MENU SCHWINGUNG
1940 !
1950 RESTORE 1960 :: CALL WR
ITE(1,15,3)
1960 DATA SCHWINGUNGSLEHRE,,
,1> FADENPENDEL,,2> FEDERPEN
DEL,,3> HAUPTMENU
1970 GOSUB 2250 :: CALL DELS
PRITE(ALL):: ON K GOTO 1980,
2010,660,1970,1970,1970,1970
1980 RESTORE 1990 :: CALL WR
ITE(5,11,1):: GOSUB 2370
1990 DATA 1> DAUER D. EIGENS
CHWINGUNG,2> FREQUENZ,3> FAD
EN- O. SEILLAENGE,4> RUECKSP
RUNG
2000 GOSUB 2280 :: ON K GOTO
 2070,2090,2100,1950,2000,20
00,2000
2010 RESTORE 2020 :: CALL WR
ITE(1,15,1):: GOSUB 2370
2020 DATA 1> RICHTGROESSE D.
 FEDER,2> DAUER D. EIGENSCHW
INGUNG,"...FREQUENZ",3> GESC
HW. IM O-PUNKT
2030 DATA 4> SPANNARBEIT =>F
EDERKONST.,5> FEDERWEG IN VE
RBINDUNG,"...MIT GELEISTETER
 ENERGIE",6> RUECKSPRUNG
```

```
2040 GOSUB 2280 :: ON K GOTO
 2110,2160,2180,2220,2190,19
50,2040
2050 !
2060 ! RECHNEN SCHWINGUNGEN
2070 GOSUB 2300 :: D=G
2080 T=2*PI*SQR(S/D):: GOSUB
 2440 :: N=1/T :: DISPLAY AT
(12,7):"PRO SCHWINGUNG":"":"
":"=> DIE FREQUENZ:":"":N;"H
ERTZ" :: GOSUB 2380 :: GOTO
1950
2090 GOSUB 2300 :: N=SQR(G/S
)/(2*PI):: DISPLAY ERASE ALL
:"FREQUENZ:";N;"HERTZ" :: GO
SUB 2380 :: GOTO 1980
2100 GOSUB 2480 :: GOSUB 232
0 :: S=(G*T^2)/(4*PI*N^2)::
GOSUB 2430 :: GOSUB 2380 ::
GOTO 1980
2110 GOSUB 2400 :: DISPLAY E
RASE ALL:"WAS IST BEKANNT?":
"":"1> PERIODENDAUER":"":"2>
 FREQUENZ"
2120 GOSUB 2280 :: ON K GOTO
 2130,2140,2120,2120,2120,21
20,2120
2130 GOSUB 2320 :: D=4*PI^2*
M/T^2 :: GOTO 2150
2140 GOSUB 2480 :: D=N^2*PI^
2*4*M
2150 DISPLAY ERASE ALL:"FEDE
RKONSTANTE:":"":D;"NEWTON/ME
TER" :: GOTO 2170
2160 GOSUB 2400 :: GOSUB 249
0 :: S=M :: GOTO 2080
2170 GOSUB 2380 :: GOTO 2010
2180 GOSUB 2490 :: GOSUB 240
0 :: GOSUB 2300 :: V=S*(D/M)
^1/2 :: GOSUB 2420 :: GOTO 2
170
2190 DISPLAY ERASE ALL:"WELC
HE ENERGIE":"":"BESITZT DER
KOERPER ?":"" :: INPUT "- NE
WTON*METER:":E
2200 GOSUB 2490 :: GOSUB 240
0 :: S=(2*E/D)^1/2
2210 DISPLAY ERASE ALL:"DIE
FEDER WIRD UM":"":S;"METER":
"":"ZUSAMMENGEDRUECKT O. GED
EHNT" :: GOTO 2170
2220 DISPLAY ERASE ALL:"WELC
HE ARBEIT WURDE AN":"":"DER
FEDER VERICHTET ?":"" :: INP
UT "- NEWTON/METER:":W
2230 GOSUB 2300 :: D=2*W/S :
: GOTO 2150
2240 !
2250 ! UNTERPROGRAMME
2260 CALL HCHAR(2,5,127,26):
: DISPLAY AT(20,3):"IHRE WAH
L ?" :: CALL HCHAR(19,5,127,
23)
2270 CALL HCHAR(21,5,127,23)
:: CALL SPRITE(#1,128,5,149,
228)
2280 ACCEPT AT(20,28)VALIDAT
E("1234567")BEEP SIZE(1):K
2290 RETURN
2300 DISPLAY AT(12,1)ERASE A
LL:"BITTE STRECKE EINGEBEN":
"":"* ABSTAND":"* HOEHE":"*
RADIUS":"* DURCHMESSER":"* L
AENGE":"* USW."
2310 INPUT "- IN METER:":S :
: RETURN
2320 DISPLAY ERASE ALL:"BITT
E DIE ZEIT EINGEBEN":"" :: I
NPUT "- IN SEKUNDEN :":T ::
RETURN
2330 DISPLAY ERASE ALL:"GESC
HWINDIGKEITSEINGABE":"":"* A
UCH ANFANGSGESCHW.":"" :: IN
PUT "- IN METER/SEK. :":V ::
 RETURN
2340 DISPLAY AT(24,3):"NOCH
EINMAL ?.......<J/N>"
2350 ACCEPT VALIDATE("NJ")BE
EP SIZE(1):K$
2360 RETURN
2370 DISPLAY AT(19,1)BEEP:"W
ELCHE KENNZIFFER BITTE :" ::
 CALL HCHAR(20,29,127,3):: R
ETURN
2380 CALL HCHAR(24,31,62,2)
2390 CALL KEY(0,K,S):: IF S=
0 THEN 2390 :: CALL CLEAR ::
 RETURN
2400 DISPLAY ERASE ALL:"BITT
E DIE MASSE EINGEBEN":"" ::
INPUT "- IN KG :":M :: RETUR
N
2410 DISPLAY ERASE ALL:"DIE
BESCHLEUNIGUNG BETRAEGT?":""
:"* ODER VERZOEGERUNG":"" ::
 INPUT "- M/S% :":A :: RETUR
N
2420 VH=V*3.6 :: DISPLAY AT(
8,1)BEEP ERASE ALL:"DIE GESC
HWINDIGKEIT BETRAEGT":"":V;"
M/SEK. ..ODER":"":VH;"KM/H"
:: RETURN
2430 DISPLAY AT(8,1)BEEP ERA
SE ALL:"STRECKE,LAENGE O. HO
EHE:":"":S;"METER" :: RETURN
2440 DISPLAY AT(10,1)BEEP ER
ASE ALL:"ZEIT:";T;"SEKUNDEN"
 :: RETURN
```

```
2450 DISPLAY AT(3,8):"*  M E
 R K E  *" :: CALL HCHAR(1,1
0,42,15):: CALL HCHAR(5,10,4
2,15):: RETURN
2460 DISPLAY AT(15,1)ERASE A
LL:"(ABWURF-,NEIGUNGS-) WINK
EL ?" :: INPUT "- IN GRAD :"
:WI :: A=WI*PI/180 :: RETURN
2470 DISPLAY ERASE ALL:"GESA
MTUMDREHUNGEN ODER":"":"AUCH
 SCHWINGUNGEN?":"" :: INPUT
"ANZAHL:":GU :: RETURN
2480 DISPLAY ERASE ALL:"BITT
E DIE FREQUENZ EINGEBEN":""
:: INPUT "- UMDREHUNGEN PRO
SEK.:":N :: RETURN
2490 DISPLAY ERASE ALL:"FEDE
RKONSTANTE ?":"" :: INPUT "-
 IN NEWTON/METER:":D :: RETU
RN
2500 FOR Z=0 TO 30 STEP 6 ::
 CALL SOUND(-100,160,Z,180,Z
,220,Z,-7,Z):: NEXT Z :: RET
URN
2510 SUB BFARBE(VF,BF)
2520 FOR Z=1 TO 12 :: CALL C
OLOR(Z,VF,1):: NEXT Z :: CAL
L SCREEN(BF)
2530 SUBEND
2540 SUB AFARBE(VF,BF)
2550 FOR Z=2 TO 4 :: CALL CO
LOR(1,16,1,Z,11,1,12,11,1)::
 NEXT Z :: FOR Z=5 TO 11 ::
CALL COLOR(Z,VF,1):: NEXT Z
:: CALL SCREEN(BF):: SUBEND
2560 SUB WRITE(Z1,ZZ,SP)
2570 CALL CLEAR :: FOR Z=Z1
TO ZZ STEP 2 :: READ A$ :: D
ISPLAY AT(Z,SP):A$ :: NEXT Z
 :: SUBEND
2580 SUB CFARBE(C):: FOR Z=3
 TO 8 :: CALL COLOR(Z,16,7):
: NEXT Z :: CALL COLOR(2,16,
5,12,7,7,1,7,1,9,C,1):: SUBE
ND
```


Fortsetzung von Seite 12

dieses Zeichen dazu verwendet, die Spalten 32 – 80 auf dem Ausdruck zu füllen.
Speichern Sie dieses Unterprogramm ab Zeile 32500 mit 'MERGE' ab. Die Zeilen 310 und 320 dienen nur dazu, das Programm auszutesten und brauchen nicht mit übernommen zu werden.

V. Schirrmann

Programm:	*BCOPY*
Autor	*Volker Schirrmann, Düppelstraße 71, 4200 Oberhausen 1*
Kategorie:	*Utility*
Sprache:	*Ext.-Basic*
Ben. Zubehör:	*Konsole, XBasic-Modul RS232 Schnittstelle Seikosha GP100A*
Speicherbelegung:	*1044 Bytes*
Optimierung:	*Befriedigend*
Kurzbeschreibung:	*Programm für Bildschirmkopie wahlweise Normal- oder Breitdruck.*

```
100 ! **********************
110 ! *                    *
120 ! *       Bcopy        *
130 ! *    Copyright by    *
140 ! *    V.Schirrmann    *
150 ! *                    *
160 ! *   Bearbeitet vom   *
170 ! *      Team des      *
180 ! *  Aktuell Verlages  *
190 ! *      Muenchen      *
200 ! *                    *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole  *
230 ! *     Ext. Basic     *
240 ! *    Schnittstelle   *
250 ! *   Drucker(GP100A)  *
260 ! *                    *
270 ! *  Speicherbelegung  *
280 ! *     1044 Bytes     *
290 ! **********************
295 !
300 CALL CLEAR :: FOR N=1 TO
 30 :: CALL HCHAR(INT(RND*23
)+1,INT(RND*31)+1,INT(RND*96
)+32,INT(RND*31)+1):: NEXT N
310 CALL BCOPY
32500 SUB BCOPY :: CALL SOUN
D(-100,-3,0)
32510 CALL KEY(0,K,S):: IF K
=74 THEN 32520 ELSE IF K=78
THEN 32600 ELSE 32510
32520 OPEN #1:"PIO" :: A$=RP
T$("-",40):: B$="-----------
BILDSCHIRMAUSDRUCK----------
-" :: PRINT #1:CHR$(14);A$;B
$;A$
32524 CALL SOUND(-300,-1,0)
32525 CALL KEY(0,K,S):: IF S
=0 THEN 32525 :: FUELL=K
```

```
32530 FOR ZEI=1 TO 24 :: FOR
 SPA=1 TO 32
32540 CALL GCHAR(ZEI,SPA,ZCH
)
32550 PRINT$=PRINT$&CHR$(ZCH
):: IF SPA=32 THEN 32560 ELS
E 32570
32560 PRINT$=PRINT$&RPT$(CHR
$(FUELL),8):: PRINT #1:PRINT
$ :: PRINT$=""
32570 NEXT SPA :: NEXT ZEI
32580 PRINT #1:A$;CHR$(10);C
HR$(10);CHR$(10)
32590 SUBEXIT
32600 OPEN #1:"PIO" :: A$=RP
T$("-",80)
32610 B$="-----------------
------------BILDSCHIRMAUSDR
UCK-----------------------
------"
32620 PRINT #1:CHR$(15);A$;B
$;A$
32624 CALL SOUND(-300,-1,0)
32625 CALL KEY(0,K,S):: IF S
=0 THEN 32625 :: FUELL=K
32630 FOR ZEI=1 TO 24 :: FOR
 SPA=1 TO 32
32640 CALL GCHAR(ZEI,SPA,ZCH
)
32650 PRINT$=PRINT$&CHR$(ZCH
):: IF SPA=32 THEN 32660 ELS
E 32670
32660 PRINT$=PRINT$&RPT$(CHR
$(FUELL),48):: PRINT #1:PRIN
T$ :: PRINT$=""
32670 NEXT SPA :: NEXT ZEI
32680 PRINT #1:A$ :: PRINT #
1:CHR$(10);CHR$(10);CHR$(10)
32690 SUBEND
```

FLUGTAG

In einer deutschen Kleinstadt findet ein Wettbewerb im Fesselballonfliegen statt. Sieger ist, wer die meisten Starts und Landungen fehlerfrei ausführen kann. Jeder Teilnehmer darf zwei Reserveballons bereithalten. Landungen werden nur anerkannt, wenn sie genau auf dem Startplatz stattfinden. Der Aufstieg erfolgt durch Ballastabwurf, die Landung durch Gasablassen. Die Flugrichtung wird durch die verschieden starken Höhenwinde beeinflußt. Links und rechts wird das Flugfeld mit Fesselballons eingegrenzt. Achten Sie darauf, daß Sie nicht in die Leinen dieser Ballons geraten, oder mit dem Beobachtungsflugzeug zusammenstoßen. In diesen Fällen ist der Flugtag vorzeitig beendet. Bemerken Sie rechtzeitig, daß Sie die Kontrolle über den Ballon verlieren, können Sie sich mit einem Fallschirm retten. Der Ballon geht dann zwar verloren, aber Ihr Leben ist gerettet und Sie dürfen mit einem Reserveballon weiter am Flugtag teilnehmen. *Marcus Gröber*

Programm: *Flugtag*
Autor: *Marcus Gröber, Dresdnerstr. 2 5090 Leverkusen*
Kategorie: *Spiele*
Sprache: *Ext.-Basic*
Ben. Zubehör: *Konsole, XBasic-Modul optional: Speech-Synthesizer Joystick (1)*
Speicherbelegung: *12141 Bytes*
Optimierung: *Gut*
Kurzbeschreibung: *Mit einem Fesselballon sind möglichst viele Starts und Landungen fehlerfrei zu bewerkstelligen. Eingebaute Schwierigkeiten machen dies nicht immer einfach.*
Bemerkung: *Bei diesem Spiel wurde viel Wert auf Details gelegt. Und gerade diese kleinen Details machen das an sich einfache Programm zu einem lustigen und interessanten Spiel. Verfügt man nicht über einen Joystick oder einen Sprach-Synthesizer, ist dies für das Spiel fast ohne Bedeutung. Es läßt sich genauso gut über die Tastatur steuern.*
In einem Vorprogramm wird festgelegt, welches Zubehör man einsetzen möchte. Entscheidet man sich für die Tastatur, kann man sich anschließend aussuchen, welche Tasten man für die Steuerung benutzen möchte.

STATISTIK

Mit diesem Programm erhalten Sie ein nützliches Instrument für Ihre statistischen Berechnungen. Nachdem Sie im Hauptmenü das gewünschte Teilprogramm eingegeben haben, können Sie im Dialog mit dem Rechner Ihre Berechnungen durchführen.
Auswahl 1 im Menü gibt Ihnen eine Demonstration der 3-D Säulengrafik.
Auswahl 2 bietet Ihnen die Möglichkeit, prozentuale oder reelle Werte für bis zu 6 Säulen einzugeben.
Mit Punkt 3 im Menü berechnen Sie den Mittelwert, die Varianz und die Standardabweichung der zuvor von Ihnen eingegebenen Werte.
Im 4. Teil des Menüs werden für Kennzahlen zwischen 0 und 255 die Häufigkeitsverteilung, Summe, Quadratsumme, das Mittel, Standardfehler und noch einige statistische Berechnungen mehr durchgeführt.
Auswahl 5 des Menüs beendet das Programm.
Udo Wenzel

Programm: *Statistik*
Autor: *Udo Wenzel, 5090 Leverkusen 1, Overfeldweg 76*
Kategorie: *Anwenderprogramm*
Sprache *TI-Basic*
Ben. Zubehör: *Konsole*
Speicherbelegung: *5778 Bytes*
Optimierung: *Befriedigend*
Kurzbeschreibung: *Das Programm erstellt Säulendiagramme und führt statistische Berechnungen in der Art des Datenverwaltungs- und Analysemoduls durch.*
Bemerkung: *Säulendiagramme gibt es mittlerweile fast ebenso viele, wie Pac-Man-Versionen. Dieses Programm kann jedoch wesentlich mehr als nur Säulen zu erstellen. Mittelwert, Standardabweichung, Standardfehler und Häufigkeitsverteilung sind nur ein Teil der Berechnungen, die hier ausgeführt werden.*
Für jeden, der mit statistischen Berechnungen zu tun hat, kann dieses Programm zu einem nützlichen Instrument werden.

```
100 ! **********************
110 ! *                    *
120 ! *      FLUGTAG       *
130 ! *    Copyright by    *
140 ! *   Marcus Groeber   *
150 ! *                    *
160 ! *   Bearbeitet vom   *
170 ! *      Team des      *
180 ! *  Aktuell Verlages  *
190 ! *      Muenchen      *
200 ! *                    *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole  *
230 ! *     Ext. Basic     *
240 ! * ÄJoystick(1) opt.ü *
250 ! *ÄSprach-Synth. opt.ü*
260 ! *                    *
270 ! *  Speicherbelegung  *
280 ! *    12141 Bytes     *
290 ! **********************
295 !
300 CALL CLEAR :: CALL MAGNI
FY(3)
310 RANDOMIZE :: DIM WI(9)
320 CALL SCREEN(5):: FOR A=0
 TO 8 :: CALL COLOR(A,2,8)::
 NEXT A
330 CALL COLOR(11,9,8,12,13,
8,13,2,15)
340 FOR A=65 TO 90 :: CALL C
HARPAT(A+32,A$):: CALL CHAR(
A,"FF"&SEG$(A$,5,12)&"FF")::
 NEXT A
350 CALL CHAR(60,"0103050911
2141FF",62,RPT$("80",8))
360 CALL CHAR(96,"00030F1F3F
383D3D3F1F0F070302010100C0F0
F8FCFCBCBCBCF8F0E0C0408080")
370 CALL CHAR(64,"000000EC92
929292")
380 FLGL$="000000000060E0602
02070FBACDFDF"&RPT$("0",24)&
"FF03FFFE" :: CALL CHAR(100,
FLGL$)
390 FLGR$="0000000000303830 2
02070FBACDFDF"&RPT$("0",24)&
"FF03FFFE"
400 CALL CHAR(108,"0000080D
FFFF0F5F5F7FFCF8000000000000
000F8FC465F5746FCF8E0E0")
410 CALL CHAR(104,RPT$("0",2
0)&"2A1C0A091401"&RPT$("0",2
0)&"150EA4CA8040")
420 CH$="0102040810204080"
430 FOR A=1 TO 8 :: CALL CHA
R(111+A,RPT$(SEG$(CH$,A*2-1,
2),8)):: NEXT A
440 CALL CHAR(91,"80F0F8DBFB
737FFF"&RPT$("0",14)&"FF010F
1FDBDFCECEFF00003C183C3C3C7E
")
450 CALL CHAR(120,RFT$("0",1
6)&"003C7E7E3C18181800183C3C
7E3C7E18183C7E7E7E3C1818")
460 CALL CHAR(124,"000000000
F95A4FFBF9F0C1C080000000000000
0000183FFFFFFC702")
470 CALL CHAR(128,"0000013DF
FFFFFFF7F7F21000000000000000C
0E0F8FCFEFEFEFEFCF8")
480 CALL CHAR(132,"60FF00000
00000F0F00000000060FF")
490 CALL CHAR(32,RPT$("00",1
5)&"01"&RPT$("00",16))
500 CALL CHAR(36,"0000000000
000000000103050F3F3F0C000000
0000000000040F8FCBC1CBCFC18")
510 CALL CHAR(136,"00924A000
0EEA8EC888E00004A92000000495
2000097D4B5959700005249")
520 CALL CHAR(140,RPT$("00",
9)&"070E0805030102"&RPT$("00
",9)&"C0602040800080"):: GOS
UB 1370
530 CALL HCHAR(1,1,120,608):
: CALL VCHAR(20,3,91):: CALL
 VCHAR(20,31,93):: FOR A=0 T
O 8 :: CALL COLOR(A,2,4):: N
EXT A
540 CALL SPRITE(#16,128,16,3
,100,0,4,#17,128,16,8,30,0,5
,#18,128,16,22,190,0,6,#19,1
28,16,3,2,0,7)
550 CALL HCHAR(22,1,132,32):
: CALL HCHAR(23,1,133,32)
560 X=3 :: C=119 :: FOR Y=19
 TO 7 STEP -2 :: CALL VCHAR(
Y-1,X,C,2):: CALL VCHAR(Y-1,
34-X,231-C,2):: C=C-1 :: IF
C=111 THEN C=119 :: X=X+1
570 NEXT Y :: CALL SPRITE(#3
,108,7,30,14,#4,108,7,30,233
)
580 CALL SPRITE(#1,124,7,168
,235)
590 FOR X=4 TO 30 :: IF RND<
.5 THEN CALL VCHAR(19,X,121+
INT(RND*3))
600 NEXT X :: CALL MOTION(#1
,0,-10)
610 FOR A=30 TO 14 STEP -1 :
: CALL SOUND(-999,-3,A):: NE
XT A
```

```
620 CALL SOUND(-999,-3,14)::
 CALL POSITION(#1,Y,X):: IF
X>120 THEN 620 ELSE CALL MOT
ION(#1,-5,-10)
630 CALL SOUND(-999,-3,8)::
CALL POSITION(#1,Y,X):: IF Y
>18 THEN 630 ELSE CALL MOTIO
N(#1,0,-10)
640 FOR A=8 TO 30 :: CALL SO
UND(-999,-3,A):: NEXT A
650 HA=INT(RND*50+95):: CALL
 SPRITE(#10,100,2,153,HA)
660 FOR A=0 TO 9 :: WI(A)=IN
T(RND*14-7):: IF WI(A)=0 THE
N WI(A)=1
670 NEXT A :: GOSUB 2050 ::
BAL,PE,LA,LV,SC=0 :: SS=15
680 BAL=BAL+1 :: BCO=BAL*3
690 L=30 :: CALL SPRITE(#2,9
6,BCO,150,240,0,-5,#13,104,1
6,152,237,0,-5)
700 CALL POSITION(#2,Y,X)::
IF X>HA+4 THEN CALL SOUND(1,
-3,L):: L=40-L :: GOTO 700 E
LSE CALL MOTION(#2,0,0,#13,0
,-7)
710 CALL POSITION(#13,Y,X)::
 IF X>HA-18 THEN CALL SOUND(
1,-3,L):: L=40-L :: GOTO 710
 ELSE CALL MOTION(#13,0,0)
720 SND$=RPT$("^",SS):: COL=
1 :: CO2=BCO+1 :: DISPLAY AT
(24,1):"HOCH ZUM START" :: C
ALL TALK(VSTART$)
730 CALL KEY(0,K,S):: CALL J
OYST(1,X,Y):: FOR D=1 TO 40
:: NEXT D :: IF Y<4 AND K<>A
SC(UP$)THEN CALL COLOR(#2,CO
L):: COL=CO2-COL :: GOTO 730
740 CALL COLOR(#2,BCO):: YD=
-1.5 :: DISPLAY AT(24,1):SND
$;TAB(18);"ALT.: 0§" :: ALM=
0
750 IF RND>.02 THEN 770 ELSE
 H=INT(RND*10):: WI(H)=INT(R
ND*14-7)*1.1^(LAND/2):: WI(H
)=MAX(-16,MIN(16,WI(H)))
760 DISPLAY AT(24,1)SIZE(17)
:"WIND DREHT (";STR$((9-H)*8
0);"§)" :: GOSUB 2050
770 CALL POSITION(#2,Y,X)::
AL=(150-Y)*5 :: DISPLAY AT(2
4,24):STR$(MAX(0,AL));"§" ::
 ALM=MAX(AL,ALM)
780 CALL MOTION(#2,YD,WI(INT
(Y/16))):: CALL POSITION(#2,
Y,X):: IF Y<16 THEN CALL MOT
ION(#2,0,0):: YD=0
790 IF Y<151 THEN 810
800 CALL COINC(#2,#10,18,C):
: IF C=0 THEN CALL SPRITE(#1
1,140,BCO,Y,X):: GOSUB 2080
:: FU=1 :: GOTO 990 ELSE 121
0
810 CALL COINC(#1,#2,16,CO):
: IF X<30 OR X>218 OR CO THE
N CALL SPRITE(#11,140,BCO,Y,
X):: GOSUB 2080 :: GOTO 970
820 IF JT$="J" THEN 860
830 CALL KEY(0,K,S):: IF S=0
 THEN 750 ELSE IF K=ASC(AS$)
THEN 910
840 IF K=ASC(DW$)THEN YY=-4
:: GOTO 880
850 IF K=ASC(UP$)THEN 890 EL
SE 750
860 CALL KEY(1,K,S):: IF S<>
0 THEN 910
870 CALL JOYST(1,XX,YY):: IF
 YY=0 THEN 750
880 IF YY=-4 THEN YD=MIN(YD+
1.5,6):: CALL SOUND(300,-5,8
):: GOTO 750
890 IF SND$=RPT$(" ",SS)THEN
 750 ELSE SND$=SEG$(SND$&" "
,2,SS):: DISPLAY AT(24,1)SIZ
E(17):SND$ :: CALL SPRITE(#1
1,32,2,Y,X,25,0)
900 CALL POSITION(#11,Y,X)::
 IF Y<144 THEN 900 ELSE CALL
 DELSPRITE(#11):: CALL SOUND
(1,-7,6):: YD=MAX(YD-1.5,-6)
:: GOTO 750
910 CALL SPRITE(#11,140,BCO,
Y,X,5,-2-(X<128)*4):: DISPLA
Y AT(24,1):"** AUSGESTIEGEN
**" :: FU,BD=0 :: YD=-6 :: S
C=SC-100
920 IF BD=1 THEN 950
930 CALL POSITION(#2,Y,X)::
CALL MOTION(#2,YD,WI(INT(Y/1
6))):: IF Y<10 THEN CALL MOT
ION(#2,0,0):: YD=0
940 CALL COINC(#1,#2,16,CO):
: IF X<30 OR X>218 OR CO THE
N GOSUB 2080 :: BD=1
950 CALL POSITION(#11,Y,X)::
 IF Y>144 THEN CALL MOTION(#
11,0,0):: FU=1
960 IF FU=0 OR BD=0 THEN 920
 ELSE 990
970 XD=-2-(X<128)*4 :: CALL
MOTION(#11,5,XD)
980 CALL POSITION(#11,Y,X)::
 IF Y<145 THEN 980 ELSE CALL
 MOTION(#11,0,0):: GOTO 990
```

```
990 CALL TALK(VEND$):: CALL
SPRITE(#12,36,16,145,240,0,-
10):: N=300
1000 CALL POSITION(#12,YY,XX
):: IF XX>X THEN CALL SOUND(
-500,N,7,N+100,10):: N=700-N
 :: GOTO 1000 ELSE CALL MOTI
ON(#12,0,0)
1010 FOR A=1 TO 8 :: CALL SO
UND(-1,INT(RND*3-7),INT(RND*
15+5)):: NEXT A
1020 CALL DELSPRITE(#11):: C
ALL MOTION(#12,0,-14):: N=30
0
1030 CALL POSITION(#12,YY,XX
):: IF XX>8 THEN CALL SOUND(
-500,N,7,N+100,10):: N=700-N
 :: GOTO 1030 ELSE CALL DELS
PRITE(#12)
1040 IF BAL<3 AND FU=1 THEN
L=30 :: CALL MOTION(#13,0,7)
:: FU=0 ELSE 1070
1050 CALL POSITION(#13,Y,X):
: IF X<235 THEN CALL SOUND(1
,-3,L):: L=40-L :: GOTO 1050
 ELSE CALL MOTION(#13,0,0)
1060 SC=SC-100 :: GOTO 680
1070 FOR A=120 TO 660 STEP 2
0 :: CALL SOUND(-100,A,6)::
NEXT A
1080 DISPLAY AT(6,7)SIZE(15)
:")) GAME OVER (("
1090 IF FU=0 THEN DISPLAY AT
(8,3)SIZE(24):"DU LIEGST IM
KRANKENHAUS" :: SC=SC-200 EL
SE DISPLAY AT(8,3)SIZE(21):"
KEINE BALLONS MEHR DA"
1100 GL$="GUTE LANDUNGEN:"&S
TR$(LA):: VP$="UND VERPATZT
 :"&STR$(PE):: DISPLAY AT(11
,5)SIZE(LEN(GL$)):GL$ :: DIS
PLAY AT(13,5)SIZE(LEN(VP$)):
VP$
1110 SC$=STR$(SC)&" PUNKTE"
:: DISPLAY AT(16,14-LEN(SC$)
/2)SIZE(LEN(SC$)):SC$
1120 DISPLAY AT(18,7)SIZE(16
):"NOCHMAL (J/N)?"
1130 CALL KEY(3,K,S):: IF K<
>74 AND K<>78 THEN 1130 ELSE
 DISPLAY AT(18,22)SIZE(1):CH
R$(K)
1140 FOR A=6 TO 18 :: CALL H
CHAR(A,4,120,25):: NEXT A
1150 IF K=74 OR K=106 THEN 6
50
1160 DISPLAY AT(12,3)SIZE(23
):"ADIOS,TSCHUESS,GOOD BYE"
1170 GOSUB 2000
1180 FOR D=1 TO 1000 :: NEXT
 D
1190 CALL DELSPRITE(ALL):: C
ALL CLEAR
1200 END
1210 CALL MOTION(#2,0,0):: L
A=LA+1 :: LV=1-LV :: IF LV=1
 AND SS>5 THEN SS=SS-1 :: DI
SPLAY AT(24,1):"EIN SANDSACK
 WENIGER." :: GOTO 1240
1220 DISPLAY AT(24,1):"TOWER
:MEHR WIND!" :: FOR A=0 TO 9
 :: IF WI(A)>-16 AND WI(A)<1
6 THEN WI(A)=WI(A)*1.1
1230 NEXT A
1240 CALL TALK(VDANG$):: SC=
SC+ALM*5+100*LA :: CALL LOCA
TE(#2,150,HA+4)
1250 CALL POSITION(#13,Y,X):
: FOR A=1 TO 15 :: CALL SOUN
D(1,-5,5):: CALL LOCATE(#13,
Y-5,X):: FOR D=1 TO 70 :: NE
XT D :: CALL LOCATE(#13,Y,X)
1260 NEXT A :: IF LA/3<>INT(
LA/3)THEN 720 ELSE DISPLAY A
T(24,1):"DEINE ";STR$(LA);".
 LANDUNG HIER!" :: SC=SC+200
1270 CALL TALK(VGOOD$)
1280 CALL POSITION(#1,Y,X)::
 IF X>146 OR X<110 THEN 1280
1290 FOR A=14 TO 16 :: CALL
POSITION(#1,Y,X):: CALL SOUN
D(1,110,5,-5,4):: CALL SPRIT
E(#A,140,3+(A-14)*2,Y,X,5,0)
:: FOR D=1 TO 20 :: NEXT D :
: NEXT A
1300 IF BAL>1 THEN DISPLAY A
T(24,1):"EIN ZUSATZBALLON!"
:: BAL=BAL-1
1310 FOR A=1 TO 3
1320 CALL POSITION(#A+13,Y,X
):: IF Y<150 THEN 1320 ELSE
CALL MOTION(#A+13,0,0)
1330 CALL SOUND(1,-6,2):: NE
XT A :: CALL DELSPRITE(#14,#
15,#16):: CALL MOTION(#13,0,
7)
1340 CALL POSITION(#13,Y,X):
: IF X<HA-4 THEN CALL SOUND(
1,-3,L):: L=40-L :: GOTO 134
0 ELSE CALL MOTION(#13,0,5,#
2,0,5)
```

```
1350 CALL POSITION(#2,Y,X)::
 IF X<240 THEN CALL SOUND(1,
-3,L):: L=40-L :: GOTO 1350
ELSE CALL MOTION(#13,0,0)::
CALL DELSPRITE(#2)
1360 BAL=BAL-1 :: GOTO 680
1370 CALL HCHAR(1,1,120,768)
:: CALL HCHAR(24,1,92,32)
1380 CALL CHAR(32,"FF0000000
00000FF")
1390 FOR A=1 TO 8 :: CALL VC
HAR(A,3,111+A):: CALL VCHAR(
A,31,120-A):: NEXT A :: FOR
A=9 TO 16 :: CALL VCHAR(A,3,
128-A):: CALL VCHAR(A,31,103
+A):: NEXT A
1400 FOR A=17 TO 23 :: CALL
VCHAR(A,3,95+A):: CALL VCHAR
(A,31,136-A):: NEXT A
1410 CALL VCHAR(24,3,91):: C
ALL VCHAR(24,31,93)
1420 CALL SPRITE(#16,128,16,
3,100,0,4,#17,128,16,8,30,0,
5,#18,128,16,22,190,0,6,#19,
128,16,3,2,0,7)
1430 DISPLAY AT(6,8)SIZE(12)
:"öβö<FLUGTAG>" :: DISPLAY A
T(7,8)SIZE(2):"ü";CHR$(127)
1440 DISPLAY AT(9,6)SIZE(18)
:"öβö<AUF UND DAVON>" :: DIS
PLAY AT(10,6)SIZE(2):"ü";CHR
$(127)
1450 DISPLAY AT(12,6)SIZE(18
):"öβö<IM LUFTBALLON>" :: DI
SPLAY AT(13,6)SIZE(2):"ü";CH
R$(127)
1460 DISPLAY AT(19,5)SIZE(20
):"öβö<ZUM START TASTE>" ::
DISPLAY AT(20,5)SIZE(2):"ü";
CHR$(127)
1470 CALL SPRITE(#4,100,3,17
7,103,#5,100,6,177,120,#6,10
0,9,177,137)
1480 CALL SPRITE(#1,96,3,175
,107,0,0,#2,96,6,175,124,0,0
,#3,96,9,175,141,0,0):: FOR
D=1 TO 50
1490 CALL KEY(0,K,S):: IF S=
1 THEN 1630
1500 NEXT D
1510 FOR A=1 TO 3 :: CALL MO
TION(#A,-2,0):: CALL SOUND(1
0,-7,3):: FOR D=1 TO 30+A*20
 :: NEXT D :: NEXT A
1520 CALL MOTION(#INT(RND*3+
1),-4,INT(RND*10-5))
1530 FOR A=1 TO 3 :: CALL PO
SITION(#A,Y,X):: IF Y<20 THE
N CALL MOTION(#A,0,0):: CALL
 COLOR(#A,7):: CALL PATTERN(
#A,136):: PENG=PENG+1 :: CAL
L SOUND(200,-7,3)
1540 IF X<30 THEN CALL MOTIO
N(#A,-4,5)ELSE IF X>215 THEN
 CALL MOTION(#A,-4,-5)
1550 CALL KEY(0,K,S):: IF S=
1 THEN 1630
1560 IF PENG=3 THEN 1580
1570 NEXT A :: GOTO 1520
1580 FOR A=1 TO 3 :: CALL PA
TTERN(#A,140):: CALL COLOR(#
A,A*3):: CALL MOTION(#A,5,0)
:: NEXT A
1590 CALL MOTION(#INT(RND*3+
1),5,INT(RND*10-5)):: CALL K
EY(0,K,S):: IF S=1 THEN 1630
1600 FOR A=1 TO 3 :: CALL PO
SITION(#A,Y,X):: IF Y>165 TH
EN CALL MOTION(#A,0,0):: CAL
L SOUND(10,-3,5):: CALL DELS
PRITE(#A):: PENG=PENG-1 :: I
F PENG=0 THEN 1480
1610 IF X<30 THEN CALL MOTIO
N(#A,4,5)ELSE IF X>215 THEN
CALL MOTION(#A,4,-5)
1620 NEXT A :: CALL KEY(0,K,
S):: IF S<1 THEN 1590
1630 CALL DELSPRITE(ALL):: C
ALL CHAR(32,""):: CALL CLEAR
1640 FOR A=65 TO 90 :: CALL
CHARPAT(A,A$):: CALL CHAR(A,
"00"&SEG$(A$,3,12)&"00"):: N
EXT A
1650 DISPLAY AT(6,11):"FLUGT
AG" :: DISPLAY AT(7,11):"---
----" :: DISPLAY AT(9,1):"(C
) BY MG-SOFTWARE 1984"
1660 DISPLAY AT(11,1):"GEWID
MET DEM LSC LEVERKUSEN"
1670 DISPLAY AT(14,1):"SPEEC
H-SYNTHESIZER (J/N)? J" :: D
ISPLAY AT(16,1):"ANLEITUNG N
OETIG (J/N)? N"
1680 DISPLAY AT(23,2):"BITTE
 ALPHA LOCK EINRASTEN"
1690 ACCEPT AT(14,27)VALIDAT
E("JN")SIZE(-1):SP$ :: IF SP
$="" THEN 1690
1700 ACCEPT AT(16,25)VALIDAT
E("JN")SIZE(-1):IN$ :: IF IN
$="" THEN 1700
1710 IF SP$="N" THEN 1730
```

```
1720 VDANG$="LOOK OUT" :: VS
TART$="PRESS UP TO START" ::
 VEND$="I AM SORRY BUT YOU W
ERE NOT GOOD" :: VGOOD$="YOU
 WERE VERY GOOD"
1730 CALL CLEAR :: IF IN$="J
" THEN GOSUB 1860
1740 DISPLAY AT(11,1):"HOCH.
....--)" :: DISPLAY AT(13,1)
:"RUNTER...--)" :: DISPLAY A
T(15,1):"ABSPRUNG --)"
1750 DISPLAY AT(6,1):"(J)OYS
TICK / (T)ASTATUR? -)T" :: A
CCEPT AT(6,28)SIZE(-1)VALIDA
TE("JT"):JT$ :: IF JT$="" TH
EN 1750
1760 IF JT$="J" THEN DISPLAY
 AT(8,2):"BITTE ALPHA LOCK L
OESEN" ELSE DISPLAY AT(8,2):
"BITTE ALPHA LOCK EINRASTEN"
1770 IF JT$="T" THEN 1800
1780 UP$=" "
1790 DISPLAY AT(11,13):"J1 H
OCH" :: DISPLAY AT(13,13):"J
1 RUNTER" :: DISPLAY AT(15,1
3):"FIRE (J1)" :: GOTO 1840
1800 FOR A=11 TO 15 STEP 2 :
: DISPLAY AT(A,13):: NEXT A
1810 VA$="=/,. " :: ACCEPT A
T(11,13)SIZE(1)VALIDATE(UALP
HA,NUMERIC,VA$):UP$ :: IF UP
$="" THEN UP$=" "
1820 ACCEPT AT(13,13)SIZE(1)
VALIDATE(UALPHA,NUMERIC,VA$)
:DW$ :: IF DW$="" THEN DW$="
 "
1830 ACCEPT AT(15,13)SIZE(1)
VALIDATE(UALPHA,NUMERIC,VA$)
:AS$ :: IF AS$="" THEN AS$="
 "
1840 DISPLAY AT(20,1):"(S)PI
EL/(A)ENDERN? S" :: ACCEPT A
T(20,20)VALIDATE("SA")SIZE(-
1):A$ :: IF A$="A" THEN 1750
1850 CALL CLEAR :: RETURN
1860 PRINT TAB(11);"FLUGTAG"
: :
1870 PRINT "ZIEL VON *FLUGTA
G* IST ES,..MIT EINEM BALLON
 MOEGLICHST OFT HOCH AUFZUST
EIGEN.DAZU..STEHEN DREI BALL
ONS ZUR VER-"
1880 PRINT "FUEGUNG.SIE UNTE
RSCHEIDEN   SICH IN DER FARB
E.GESTEUERT WIRD DER BALLON
DURCH GASAB-LASSEN (SINKEN)U
ND BALLAST-"
1890 PRINT "ABWURF (STEIGEN)
.DER BALLON WIRD DURCH WINDE
 HIN-UND....HERGERISSEN."
1900 PRINT "VORSICHT VOR DEM
 SPORTFLUG- ZEUG UND DEN FES
SELBALLON-..LEINEN."
1910 PRINT : :TAB(25);")))"
1920 CALL KEY(0,K,S):: IF S<
1 THEN 1920 ELSE CALL CLEAR
1930 PRINT "DIE LANDUNG MUSS
 GENAU AUF..DEM STARTPLATZ E
RFOLGEN.....SONST VERLIERT M
AN EINEN....BALLON.STOESST M
AN GEGEN DAS"
1940 PRINT "FLUGZEUG ODER DI
E LEINEN,SO IST SPIELENDE,AU
SSER,MAN HATSICH DURCH EINEN
 ABSPRUNG"
1950 PRINT "GERETTET.DER BAL
LON ZER-....SCHELLT DANN ZWA
R,DAS SPIEL GEHT ABER WEITER
.IN DER UN- TERSTEN ZEILE WE
RDEN HOEHE"
1960 PRINT "UND BALLASTMENGE
 ANGEZEIGT. ALLE DREI LANDUN
GEN GIBT ES EINEN ZUSATZBALL
ON."
1970 PRINT :TAB(25);")))": :
 :
1980 CALL KEY(0,K,S):: IF S<
1 THEN 1980 ELSE CALL CLEAR
:: RETURN
1990 GOTO 1990
2000 CALL POSITION(#1,Y,X)::
 IF X<200 OR X>220 THEN 2000
 ELSE CALL MOTION(#1,11,-10)
2010 CALL SOUND(-999,-3,14):
: CALL POSITION(#1,Y,X):: IF
 Y<166 THEN 2010 ELSE CALL M
OTION(#1,0,-8)
2020 CALL SOUND(-999,-3,20):
: CALL POSITION(#1,Y,X):: IF
 X>20 THEN 2020 ELSE CALL MO
TION(#1,0,0)
2030 FOR A=20 TO 30 :: CALL
SOUND(-999,-3,A):: NEXT A
2040 RETURN
2050 IF WI(9)<1 THEN CALL CH
AR(100,FLGL$)ELSE CALL CHAR(
100,FLGR$)
2060 FOR W=16 TO 19 :: CALL
MOTION(#W,0,(W-12)*SGN(WI(0)
)):: NEXT W
2070 RETURN
```

Fortsetzung auf Seite 29

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE



KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre TI REVUE nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Euer" Kiosk nicht beliefert wurde? Kein Problem! Für ganze 30 DM liefern wir Euch per Post sechs Hefte ins Haus (Ausland 40 DM). Einfach den Bestellschein auf der nächsten Seite ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). TI REVUE kommt dann pünktlich ins Haus.

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen— Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für sechs Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

DAS GROSSE SONDERHEFT EIN MUSS FÜR JEDEN TI-USER

RUND 150 SEITEN

LISTINGS! LISTINGS! LISTINGS!



Achtung TI-User! Jetzt kommt für Sie das große Listing-Sonderheft von TI-REVUE. Rund 150 Seiten Listings. Spiele, Profiprogramme, Lehrprogramme, für Schule und Hobby. Aus diesen Listings können Sie ersehen, wie andere TI-User Aufgaben lösen. Ein unbedingtes Muß für jeden programmierenden TI-Besitzer. Dazu als Sonderservice: Eine Übersicht aller Firmen, die noch TI-Hard- und Softwaren führen! Der große TI-Katalog. Nur 12,80 DM!
Erscheinung: Am 29. Mai. Sie erhalten das Heft bis zu einer Woche früher!
Reservieren Sie sich Ihr Exemplar mit dem Coupon auf der nächsten Seite. Achtung: Kein Geld senden, wir schicken per Nachnahme oder gegen Vorausrechnung bei Erscheinen!

RESERVIERUNGS-KARTE

Bitte reservieren Sie mir (   ) Exemplar(e)
TI REVUE SPECIAL Nr 1/85 zum Preis von DM 12,80

Ich zahle
( ) per Nachnahme bei Erhalt (plus Postgebühr)
( ) per Vorausrechnung (Versand nach Geldeingang)
( ) per Bankabbuchung am Versandtag

Kto-Nr.: ________ Bankleitzahl ________
Bei (Bank und Ort) ________
Name ________
Straße/Hausnr. ________
Ort/PLZ ________
Unterschrift ________

Ausschneiden und einsenden an
TI-Service, Pf 1107, 8044 Lohhof

DAS SUPER-SOND PRIVATE KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen ab sofort die TI-Revue: KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!

Wir veröffentlichen nur Kleinanzeigen privater Inserenten, keine gewerblichen Anzeigen. Die kosten pro Millimeter DM 2,50 plus Mehrwertsteuer!

ABO SERVICE-KARTE

5
TI

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

TI REVUE

Abo-Service 5
Postfach 1107
8044 UNTERSCHLEISSHEIM

Coupon

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten sechs Ausgaben an untenstehende Anschrift.

Name ________
Vorname ________
Straße/Hausnr. ________
Plz/Ort ________

Ich bezahle:
☐ per beiliegendem Verrechnungsscheck
☐ gegen Rechnung
☐ bargeldlos per Bankeinzug von meinem Konto

bei (Bank) und Ort ________
Kontonummer ________
Bankleitzahl ________
(steht auf jedem Kontoauszug)
Unterschrift ________

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift ________

ER- ANGEBOT:

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!

Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!

Die Insertion ist nicht vom Kauf des Heftes abhängig!

Wir behalten uns vor, Anzeigen, die nicht zum Themenkreis des Heftes — Computer — gehören, nicht abzudrucken oder sie nur insoweit zu berücksichtigen, wie es der Umfang des kostenlosen Anzeitenteils zuläßt.

Wollt Ihr nicht mehr seitenlange Listings eintippen? Dann benutzt unseren Kassetten-Service: Für ganze zehn DM — nur Vorkasse, keine Rechnung — keine Nachnahme — erhalten Sie alle angebotenen Listings des jeweiligen Heftes auf Kassette per Post ins Haus. Achtung: Bestellungen ohne Geld werden nicht bearbeitet! Lieferzeit: In der Regel zwei bis drei Wochen! Wichtig: Unsere Kassetten werden digital in einem Profistudio von einem Masterband gezogen, nicht irgendwie per Recorder runtergenudelt. Selbstverständlich volles Umtauschrecht bei Nichtfunktionieren.

KASSETTENSERVICE-KARTE

Hiermit bestelle ich alle freigegebenen Listings dieses Heftes:
zum Sonderpreis von DM 10,–
Den Betrag von DM 10,– habe ich ( ) in bar, ( ) als Scheck beigelegt
Meine Anschrift:

Name ______________________

Vorname ______________________

Straße ______________ Nr. ______

PLZ/Ort ______________________

Unterschrift ______________________

Ausschneiden und im Brief senden an:

AKTUELL TI-
KASSETTENSERVICE 5
POSTFACH 1107
8044 LOHHOF

KLEINANZEIGEN SERVICE-KARTE

COUPON

PREISLISTE FÜR GEWERBLICHE ANBIETER

Jeder Millimeter in der 42 mm breiten Anzeigenspalte kostet als Fließtext-Anzeige DM 2,50 plus MWSt. Gestaltete Anzeigen pro mm DM 3,— plus MWSt. Sie erhalten eine Rechnung.

Hiermit bestelle ich gemäß Ihrem Angebot in der nächsterreichbaren Ausgabe von TI-REVUE folgende

( ) Privatanzeige
( ) Gewerbliche Anzeige
— Zutreffendes ankreuzen —

TI REVUE
Anzeigenabteilg.
Postfach 1107
8044 Lohhof

Die große Börse für jeden Zweck in der TI REVUE. Kostenlos für Privat-Inserenten. Spottbillig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Achtung! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt nicht TI-typisch ist oder gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatanbieter: maximal acht Zeilen à 28 Anschläge.

Name ______________________ Vorname ______________________

Strasse ______________________ (PLZ) Ort ______________________

BÖRSE

Verk. TI 99/4A + Expansions-Box + Karte für Box + Extended Basic + Handbuch (deutsch) + Rec.Kabel (Orig. TI) + TI Bücher (Tips + Tricks u.a. DM 900,–, Walter Hufler, Am Geistpförtchen 2, 6 Ffm. 1, Tel. 069/287781

TI 99/4A + ExBasic + Mini-Ferns. + Cass. Rec. + 2 Joyst. (TI) + 7 Spielmodule + Progr. a. Kass. + 4 Bücher ü. TI 99/4A und mehr, zus. nur 900,–, Tel. 06431/53131

Super-Grafik-Paket mit 5 Demo (Ex-B-)Plattenzeichner (Ex-B-)Duell im All (TI-B), Moonlander (Ex-B), Drehen v. Sprites (Ex-B). Infos bei frank. Rückumschlag R. G. Watzling 10, 8250 Dorfen

Lohnsteuer sparen! LJA 1984 mit dem TI. Lst. und Est.-Tabellen und weitere Prg. Info gegen Rückporto Dieter Schiller, Rachelstr. 10a, 8400 Regensburg

99/4A kpl. m. Box + RS232 + V24 + 32K + Disk + X-Bas. + Dat Ver + Anal + Epson RX 80 m. Kabel + v. Zubeh. VB 2900,–, Tel. 06041/5285

Cassettenprogr. f. TI 99/4A, Spiele, Adventure, Nutzprogr. n. Wahl, Infoliste g. frank. Briefumschlag T. Brandau, Troppauerstr. 6, 8552 Höchstadt/A. Tel. 09193/1820

Suche TI Ext. Basic Modul bis 150,–. Tel. 0271/334867

TI-Module zu verk. Munchman 60,–, Hustle 30,–, Tombstone 35,–, Parsec 60,–, Moonsweeper 70,–. Tel. 02473/7320

Verk. TI 99/4A mit Zubehör. Tel. 08252/7986

Suche TI-Extended Basic, Beate Reischmann, Schwimmbadstr. 5, 8302 Kloten (CH)

Verk. tolle Masch.Prgr. für MM, XB u. E/A Mod. u.a. dt. Textverarb. Liste geg. DM 0,80 in Briefm. von M. Hofmann, Bauernwaldstr. 130, 7000 Stgt. 1

TI 99/4A + Ext.Basic u. Rec-Kabel + Literatur + TI. + Ex-Basic Kurs + Spielmodule + Cas. Software, 600,– DM. Tel. 05403/5675 ab 18 Uhr

Neue Mailbox! Alldas Telemod. Tel. 0243/23602

Verkaufe gebr. TI 99/4A Anlage. Auch einzeln. Information u. Liste unter der Tel. 030/4916770

TI Module günstig! Schach 65,–, PACMAN 60,–, Munchman 40,–. PARSEC 45,–, Micro-Surgeon 60,–. Tel. 0531/342633. R. Werthebach

TI 99/4A **Adressenverwalt.** Games, Utilities bei H. G. Rausch, M.-Planck-Str. 14, 53 Bonn 2, RP 2

Module Attack, Othel. Carw. je 20,–, Alpiner 30,–, Parsel 40,–, Bücher usw. Tel. 06638/1503

Suche: Diskcontroller (Ex.). Biete: Laufw. BASF-6106, SS, DD 299,– Tel. 0251/664684

Superprogramme in B. + Exb. ab 2 DM. Info g. 1 DM bei CBT-Software. Stefan Berger, Joseph-Götschstr. 8, 8093 Rott (ZB Robin Hood, Black Hole)

Verk. gegen Höehstgebot folg. Module: Statistik (engl.), Buchf., Speech Editor, Invaders, Munch Man, Video Games 2. Multipl, TI-Writer (jew. + Disk), Rechnungsst. (nur Disk), sowie alle 11 Adv. Spiele auf Cass. (Mod. erforderl.). Angebote bis 3 Wo. n. Erscheinungsdat. an M. Humbert, Saphirring 30, 3008 Garbsen 1, Versand per NN.

Editor/Ass. Anwender sucht Tauschpartner. Verkaufe int. Laufwerk 350,– DM. Hans Bednarz, Im Rundteil 13, 3055 Hagenburg, Tel. 05033/7347

Tausche Software (TI/X/EA), suche Kont. Großraum Köln. Drucke Listings geg. Unkostenerst. Suche DFÜ-Partner. B. Betz, Siegburger Str. 111, 5000 Köln 21, Tel. 0221/883105

Verkaufe TI- und Ex-Basic Programme. Info gegen Rückporto. B. Knedel. Tulpengasse 16. 3171 Weyhausen. Tel. 05362/71187

Verk. TI 99/4A P-Box RS 232, 32K Drucker Ex.Basic E/A Modul Parsec Casskabel, div. Spiele, Bücher und Literatur komplett VB 05361/74476

Verkauf: 99/4A + Box + XBasic + Disk + Control + 32K + RS232 + CassKabel + Joysticks + Minimem (+ dt. Handbuch) + Software + Literatur 2800,– DM. Markus Nikolay, Am Erzel 14, 6633 Wadgassen 3, Tel. 06834/41475

Suche TI-User zwecks Assemblerprogrammtausch. Jörg Storm, Tel. 0821/63857

Verkaufe komplette TI-Anlage in fahrb. Schrank. Dazu Bücher, 40 Disc, 20 Mod., 20 Cas. Listings usw. DM 4250,–, Tel. 02041/28321

Suche für TI 99/4A Disk-Laufwerk, Tel. 089/4703941

Suche Disk- oder Kass.-Bundesliga-Tabelle. Alwin Koop, Lagerlöfstr. 34, 2000 Hamburg 65

DIE NÄCHSTE AUSGABE ERSCHEINT AM 26. JUNI

Verk. TI 99/4A + Ext.-Basic + Reckkab. (origin. TI) + Spiele + Literatur NV 02043-71974, Thomas Fliß, August-Brust-Str. 26, 4390 Gladbeck

TI 99/4A + Extended Basic u. Buch + Kassetten + Recorder + 2 Joysticks + Othello + Rec.- und Monitorkabel. Nur alles zusammen zu verkaufen für DM 600,–, Tel. 02041/36569 nach 17 Uhr in 4250 Bottrop.

TI 99/4A + Joyst. + Pgm's u. RecKabel u.a. 180,–, Thorsten Reich, Fritz Remystr. 11, 6050 Offenbach

Verkaufe: Softwareset (Ex), 8 Programme mit Cass. gegen 10 DM-Schein. 5 Spiele + 3 prakt. Progr. z.B. Mathe, die erste 10 plus Softwarebonus. Module Attack, CAR-RACE, TOMBSTONE-CITY je 300,– DM bei V. Brose, Hamburgerstr. 1, 4750 Unna-Kgb.

TI 99/4A Datenverw. + Analyse-Modul (neu – original verpackt) 50,– DM, Tel. 02331/402220

Achtung, suche Programme für TI 99/4A. Liste senden an: Dirk Zimmer, An der alten Kantine 2, 3163 Sehnde 3.

Orig. TI 99/4A ungebr. 089/2715110, H. Ch. Steeg

Suche Music Maker-Modul, bis 35 DM, Stefan Lauterbach, Kyfitting 6, 3380 Goslar

TI-Rechner-Menügesteuert. Programm mit 14 Rechenarten. Incl. Cass. + Porto nur DM 10,– im Umschlag an: D. Karbach, Remscheider Str. 18, 5650 Solingen 1

TI 99/4A + versch. Spiel-Module + Joyst. 350,–. Tel. 06752/5735. M. Führer, 6570 Kirn

```
2080 CALL SOUND(500,-6,3)::
CALL COLOR(#2,7):: CALL MOTI
ON(#2,0,0):: CALL PATTERN(#2
,136)
2090 FOR D=1 TO 200 :: NEXT
D :: CALL DELSPRITE(#2):: DI
SPLAY AT(24,1):"..DEIN BALLO
N IST GEPLATZT."
2100 PE=PE+1 :: RETURN
2110 SUB TALK(S$)
2120 IF S$="" THEN SUBEXIT E
LSE CALL SAY(S$&",,OVER")
2130 SUBEND
```



```
100 REM*******************
110 REM*                 *
120 REM*   STATISTIK     *
130 REM* Copyright  by   *
140 REM*  Udo  Wenzel    *
150 REM*                 *
160 REM* Bearbeitet  vom *
170 REM*    Team   des   *
180 REM* Aktuell Verlages *
190 REM*     Muenchen    *
200 REM*                 *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A Konsole *
230 REM*                 *
270 REM* Speicherbelegung *
280 REM*    5778 Bytes   *
290 REM*******************
295 REM
300 CALL CLEAR
310 FOR I=1 TO 16
320 CALL COLOR(I,16,1)
330 NEXT I
340 ANZ=6
350 Z1$="FFFFFFFFFFFFFFFF"
360 A$="0103070F1F3F7FFF"
370 CALL CHAR(152,Z1$)
380 CALL CHAR(136,Z1$)
390 CALL CHAR(128,Z1$)
400 CALL CHAR(120,Z1$)
410 CALL CHAR(112,Z1$)
420 CALL CHAR(104,Z1$)
430 CALL CHAR(102,Z1$)
440 CALL CHAR(114,A$)
450 CALL CHAR(150,A$)
460 CALL CHAR(130,A$)
470 CALL CHAR(144,Z1$)
480 CALL CHAR(99,A$)
490 CALL CHAR(145,A$)
500 DIM PR(7),PP(7),RP(7),Q3
(7),Q4(7),OP(7),AS(7),ASD(7)
,FG(255),F(255),CV(100)
510 CALL COLOR(16,7,1)
520 CALL COLOR(15,10,1)
530 CALL SCREEN(2)
540 CALL CLEAR
550 PRINT TAB(5);"S T A T I
S T I K"
560 PRINT : :TAB(8);"M E N U
 E": : : :"1)-ZEICHENDEMO":
:"2)-EINGABE FUER GRAPH": :
570 PRINT "3)-STATIST 1": :"
4)-STATIST 2": :"5)-ENDE"
580 FOR I=1 TO ANZ
590 PP(I)=0
600 RP(I)=0
```

```
610 PR(I)=0
620 OP(I)=0
630 Q3(I)=0
640 Q4(I)=0
650 ASD(I)=0
660 AS(I)=0
670 NEXT I
680 CALL KEY(0,K,S)
690 IF S=0 THEN 680
700 IF K=49 THEN 760
710 IF K=50 THEN 1270
720 IF K=51 THEN 2120
730 IF K=52 THEN 2380
740 IF K=53 THEN 2970
750 GOTO 1760
760 CALL CLEAR
770 GOSUB 1800
780 CALL VCHAR(7,6,152,17)
790 CALL VCHAR(7,7,152,17)
800 CALL COLOR(16,11,1)
810 CALL VCHAR(7,8,144,17)
820 CALL HCHAR(7,7,144,2)
830 CALL HCHAR(7,6,150)
840 CALL COLOR(15,12,1)
850 CALL VCHAR(15,10,136,9)
860 CALL VCHAR(15,11,136,9)
870 CALL COLOR(14,7,1)
880 CALL COLOR(13,10,1)
890 CALL VCHAR(15,12,128,9)
900 CALL HCHAR(15,11,128,2)
910 CALL HCHAR(15,10,130)
920 CALL VCHAR(10,14,120,14)
930 CALL VCHAR(10,15,120,14)
940 CALL COLOR(12,13,1)
950 CALL COLOR(11,4,1)
960 CALL VCHAR(10,16,112,14)
970 CALL HCHAR(10,14,114)
980 CALL HCHAR(10,15,112,2)
990 CALL VCHAR(2,18,104,22)
1000 CALL VCHAR(2,19,104,22)
1010 CALL COLOR(10,15,1)
1020 CALL COLOR(9,16,1)
1030 CALL VCHAR(2,20,102,22)
1040 CALL HCHAR(2,19,102,2)
1050 CALL HCHAR(2,18,99)
1060 W$="0505 75%"
1070 GOSUB 1180
1080 W$="0913 40%"
1090 GOSUB 1180
1100 W$="1308 65%"
1110 GOSUB 1180
1120 W$="1701 100%"
1130 GOSUB 1180
1140 CALL KEY(0,K,S)
1150 IF S=0 THEN 1140
1160 CALL HCHAR(1,1,23,768)
1170 GOTO 560
1180 REM
1190 XX=VAL(SEG$(W$,1,2))
1200 YY=VAL(SEG$(W$,3,2))
1210 FOR II=1 TO LEN(W$)-4
1220 CALL HCHAR(YY,II+XX,ASC
(SEG$(W$,II+4,1)))
1230 NEXT II
1240 CALL SOUND(500,249,0,31
5,0)
1250 RETURN
1260 GOTO 560
1270 REM
1280 PRINT "WIEVIEL SAUELEN
WERDEN ES ":"SEIN?":"(HOECHS
TENS 6 STUECK)";
1290 INPUT "":ANZ
1300 IF ANZ>6 THEN 1280
1310 GOTO 1540
1320 REM
1330 PRINT "WIEVIEL VON WIEV
IEL?";
1340 FOR I=1 TO ANZ
1350 PRINT STR$(I);". EINGAB
E (XX,XX)";
1360 INPUT "":PP(I),RP(I)
1370 NEXT I
1380 PR(I)=(100*PP(I)/RP(I))
1390 GOSUB 1800
1400 GOTO 1680
1410 REM
1420 CALL CLEAR
1430 PRINT "WIEVIEL VON WIEV
IEL";
1440 FOR I=1 TO ANZ
1450 INPUT "==>":Q3(I),Q4(I)
1460 OP(I)=(Q4(I)*Q3(I))/100
1470 PR(I)=OP(I)
1480 NEXT I
1490 CALL CLEAR
1500 TY=152
1510 UYT=145
1520 RTY=144
1530 GOTO 1670
1540 CALL CLEAR
1550 PRINT "SIND DIE EINGABE
N IN PROZENT?";
1560 CALL KEY(0,K,S)
1570 IF S=0 THEN 1560
1580 IF (K=78)+(K=110)THEN 1
430
1590 FOR I=1 TO ANZ
1600 PRINT STR$(I);". EINGAB
E (%)";
```

```
1610 INPUT "":PR(I)
1620 CALL CLEAR
1630 TY=152
1640 UYT=145
1650 RTY=144
1660 NEXT I
1670 GOSUB 1800
1680 I=1
1690 PR(7)=100
1700 FOR X1=5 TO ANZ*4+5 STE
P 4
1710 PL=PR(I)
1720 L1=(PR(I)/5)+1
1730 GOSUB 1860
1740 I=I+1
1750 NEXT X1
1760 CALL KEY(0,K,S)
1770 IF S=0 THEN 1760
1780 CALL HCHAR(1,1,32,768)
1790 GOTO 560
1800 REM
1810 CALL CHAR(33,"FF")
1820 CALL CHAR(34,"181818181
8181818")
1830 CALL VCHAR(1,3,34,23)
1840 CALL HCHAR(24,3,33,29)
1850 RETURN
1860 REM
1870 REM
1880 REM
1890 Y1=24-(L1)
1900 CY=INT(Y1)-2
1910 CX=X1-1
1920 REM
1930 GOSUB 2030
1940 REM
1950 REM
1960 REM
1970 CALL VCHAR(Y1,X1,TY,L1)
1980 CALL VCHAR(Y1,X1+1,TY,L
1)
1990 CALL HCHAR(Y1-1,X1,UYT)
2000 CALL HCHAR(Y1-1,X1+1,RT
Y,2)
2010 CALL VCHAR(Y1,X1+2,RTY,
L1)
2020 RETURN
2030 W$=SEG$("0"&STR$(CX),LE
N(STR$(CX)),2)&SEG$("0"&STR$
(CY),LEN(STR$(CY)),2)&SEG$(S
TR$(PL),1,3)&"%"
2040 REM
2050 IF W$="08210%" THEN 750
2060 IF W$="12210%" THEN 750
2070 IF W$="16210%" THEN 750
2080 IF W$="20210%" THEN 750
2090 IF W$="24210%" THEN 750
2100 GOSUB 1180
2110 RETURN
2120 CALL CLEAR
2130 PRINT "ANZAHL DER EINGA
BEELEMENTE :"
2140 INPUT "==>":N
2150 PRINT : : :"EINGABEELEM
ENTE:"
2160 FOR I=1 TO N
2170 INPUT "==>":CV(I)
2180 NEXT I
2190 S=0
2200 Q=0
2210 FOR I=1 TO N
2220 S=S+CV(I)
2230 NEXT I
2240 M=S/N
2250 FOR I=1 TO N
2260 Q=Q+(CV(I)-M)^2
2270 NEXT I
2280 V=Q/(N-1)
2290 A=SQR(V)
2300 CALL CLEAR
2310 PRINT "MITTELWERT: ";M
2320 PRINT :"VARIANZ: ";V
2330 PRINT :"STANDARTABW.: "
;A
2340 PRINT : : : :"NOCH MAL
?"
2350 CALL KEY(0,KS,SK)
2360 IF SK=0 THEN 2350
2370 IF (KS=78)+(KS=110)THEN
 490 ELSE 2120
2380 PRINT : : : : : : : : :
 : : : : : : : :"BERECHNEN S
TATISTISCHER KENNZAHLEN (VON
 0 BIS 255)UND"
2385 PRINT "ZEICHNEN EINER H
AEUFIGKEITS-VERTEILUNG"
2390 S=0
2400 S2=0
2410 M0=999
2420 M9=-999
2430 H1=0
2440 H2=1
2450 INPUT "WIEVIEL WERTE ?"
:N
2460 FOR U=1 TO N
2470 INPUT "WERT :":FG(U)
2475 IF FG(U)>255 THEN 2470
2480 NEXT U
2490 FOR I=1 TO N
2500 S=S+FG(I)
```

```
2510 S2=S2+FG(I)^2
2520 F(FG(I))=F(FG(I))+1
2530 IF FG(I)>M9 THEN 2540 E
LSE 2550
2540 M9=FG(I)
2550 IF FG(I)<MO THEN 2560 E
LSE 2570
2560 MO=FG(I)
2570 NEXT I
2580 M=S/N
2590 FOR I=1 TO N
2600 H1=H1+(FG(I)-M)^2
2610 NEXT I
2620 V=H1/(N-1)
2630 A=SQR(V)
2640 FO=SQR(V/N)
2650 C=A/M
2660 PRINT :"HAEUFIGKEITSVER
TEILUNG": :
2670 FOR K=MO TO M9
2680 PRINT K,F(K),".";TAB(F(
K)+21);"*"
2690 FOR H2=H2 TO (H2+F(K)-1
)
2700 IF H2=(N+1)/2 THEN 2710
 ELSE 2720
2710 M1=K
2720 IF H2=N/2 THEN 2730 ELS
E 2740
2730 M1=K/2
2740 IF H2=(N+2)/2 THEN 2750
 ELSE 2760
2750 M1=M1+K/2
2760 NEXT H2
2770 NEXT K
2780 S9=3/A*(M-M1)
2790 PRINT :"ANZAHL SUMME QU
.SUMME MITTEL"
2800 M=INT(M*1000+.5)/1000
2810 PRINT N;TAB(8);S;TAB(15
);S2;TAB(22);M
2820 PRINT : :"MEDIAN..MAX.
..MIN. .VARIANZ"
2830 V=INT(V*1000+.5)/1000
2840 PRINT M1;TAB(8);M9;TAB(
15);MO;TAB(22);V
2850 PRINT :"ABWEI. FEHLER V
AR-K. SCHIEFE"
2860 A=INT(A*1000+.5)/1000
2870 FO=INT(FO*1000+.5)/1000
2880 S9=INT(S9*1000+.5)/1000
2890 C=INT(C*1000+.5)/1000
2900 PRINT A;TAB(8);FO;TAB(1
5);C;TAB(22);S9
2910 PRINT : : :"NOCH MAL?"
2920 S=0
2930 K=0
2940 CALL KEY(0,K,S)
2950 IF S=0 THEN 2940
2960 IF (K=78)+(K=110)THEN 5
40 ELSE 2380
2970 END
```

UNTERNEHMEN DELTA

Dieses Programm macht den TI zu einem Schießstand. Nachdem das Programm gestartet ist, bewegen sich die verschiedensten Objekte abwechselnd über den Bildschirm. Mit Hilfe des Joysticks müssen die Flugzeuge, Autos, Vögel oder Wolken mit einem Fadenkreuz eingefangen und durch Drükken des Feuerknopfes abgeschossen werden. Je weniger Zeit, desto höher das Punktekonto.

Thorsten Rauer

Programm: *Unternehmen Delta*
Autor: *Thorsten Rauer, Beetstraße 64, 4902 Bad Salzuflen 1*
Kategorie: *Spiele*
Sprache: *TI-Basic*
Ben. Zubehör: *Konsole, Joystick (1)*
Speicherbelegung: *6684 Bytes*
Optimierung: *Gut*
Kurzbeschreibung: *Über den Bildschirm bewegen sich verschiedene Objekte, die man mit einem Fadenkreuz einfangen und abschießen muß. Leicht wird einem dies nicht gemacht, die Objekte weichen ständig aus.*
Bemerkung: *Ein hervorragendes Spiel für die Grundversion. Gute Bildschirmdarstellung und rasante Geschwindigkeit.*

```
100 REM*********************
110 REM*                   *
120 REM*UNTERNEHMEN  DELTA*
130 REM*  Copyright  by    *
140 REM* Thorsten  Rauer   *
150 REM*                   *
160 REM* Bearbeitet  vom   *
170 REM*    Team   des     *
180 REM* Aktuell Verlages *
190 REM*     Muenchen      *
200 REM*                   *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole *
250 REM*   Joystick  (1)   *
260 REM*                   *
270 REM* Speicherbelegung *
280 REM*    6684 Bytes     *
290 REM*********************

295 REM
300 CALL CLEAR
310 FOR I=1 TO 12
320 CALL COLOR(I,2,I+4)
330 NEXT I
340 PRINT ".....UNTERNEHMEN
 DELTA": : : : :
350 PRINT ".....VON THORSTEN
 RAUER": : : : : : : : :
360 PRINT "....ZUM STARTEN B
ITTE DIE": :"....FEUER TASTE
 DRUECKEN "
370 CALL KEY(1,K,S)
380 IF S=0 THEN 370
390 CALL CLEAR
400 RANDOMIZE
410 CALL SCREEN(9)
420 RESTORE 1740
430 FOR AA=1 TO 24
440 READ Z,Z$
450 CALL CHAR(Z,Z$)
460 NEXT AA
470 RESTORE 1800
480 FOR AA=1 TO 9
490 READ AB,AC,AD
500 CALL COLOR(AB,AC,AD)
510 NEXT AA
520 RESTORE 1820
530 CALL CLEAR
540 CALL HCHAR(21,1,107,28)
550 CALL HCHAR(22,1,130,96)
560 FOR AA=1 TO 48
570 READ AB,AC,AD
580 CALL HCHAR(AB,AC,AD)
590 NEXT AA
600 RESTORE 1810
610 READ A,B,C,E
620 GOTO 1110
630 CALL JOYST(1,X,Y)
640 IF (X=0)*(Y=0)THEN 680
650 A=A+(Y/-4)
660 B=B+(X/4)
670 C=C-1
680 IF (A>24)+(A<1)+(B>32)+(
B<1)THEN 740
690 CALL GCHAR(A,B,XX)
700 CALL HCHAR(A,B,104)
710 CALL HCHAR(A,B,XX)
720 CALL KEY(1,K,S)
730 IF K=18 THEN 750
740 RETURN
750 IF XX=112 THEN 1300
760 IF XX=113 THEN 1280
770 IF XX=114 THEN 1320
780 IF (XX=105)+(XX=106)THEN
 1240
790 IF XX=138 THEN 1260
800 CALL HCHAR(A,B,146)
810 FOR BA=1 TO 30 STEP 5
820 CALL SOUND(-500,-5,BA)
830 CALL SOUND(-70,-6,BA-1)
840 NEXT BA
850 CALL HCHAR(A,B,XX)
860 E=E-25
870 RETURN
880 RESTORE 1870
890 GOTO 1030
900 RESTORE 1900
910 GOTO 1030
920 RESTORE 1930
930 GOTO 1030
940 RESTORE 1940
950 GOTO 1030
960 RESTORE 1980
970 GOTO 1030
980 RESTORE 2030
990 GOTO 1150
1000 RESTORE 2020
1010 GOTO 1150
1020 RESTORE 1950
1030 READ AB,AC
1040 FOR AA=1 TO AB
1050 READ AD,AE
1060 CALL HCHAR(AD,AE,AC)
1070 GOSUB 630
1080 CALL HCHAR(AD,AE,32)
1090 NEXT AA
1100 C=C-20
1110 GOSUB 1650
1120 IF C<=0 THEN 1410
1130 D=INT(RND*8+1)
1140 ON D GOTO 880,920,940,9
00,1020,960,1000,980
```

```
1150 READ AB,AC,AD,AE,G,H
1160 FOR AA=1 TO AB
1170 G=G+AD
1180 H=H+AE
1190 CALL HCHAR(G,H,AC)
1200 GOSUB 630
1210 CALL HCHAR(G,H,32)
1220 NEXT AA
1230 GOTO 1100
1240 RESTORE 2040
1250 GOTO 1330
1260 RESTORE 2050
1270 GOTO 1330
1280 RESTORE 2060
1290 GOTO 1330
1300 RESTORE 2070
1310 GOTO 1330
1320 RESTORE 2080
1330 READ AB,AC,AD,AE
1340 FOR CA=1 TO AB
1350 READ AH
1360 CALL HCHAR(A,B,AH)
1370 CALL SOUND(-450,AC,CA*A
D)
1380 NEXT CA
1390 E=E+AE-AA*10
1400 GOTO 1110
1410 CALL SOUND(500,500,0,53
5,0)
1420 RESTORE 2090
1430 FOR AA=1 TO 4
1440 READ A$,I,J
1450 FOR AB=1 TO LEN(A$)
1460 CALL HCHAR(I,J-1+AB,ASC
(SEG$(A$,AB,1)))
1470 CALL SOUND(25,110,0,101
0,0)
1480 NEXT AB
1490 NEXT AA
1500 IF E>HS THEN 1510 ELSE
1520
1510 HS=E
1520 FOR AA=1 TO LEN(STR$(HS
))
1530 CALL HCHAR(17,19+AA,ASC
(SEG$(STR$(HS),AA,1)))
1540 CALL SOUND(150,700,0,17
25,0)
1550 NEXT AA
1560 CALL KEY(0,K,S)
1570 IF S=0 THEN 1560
1580 IF K=15 THEN 300
1590 IF K=6 THEN 520
1600 IF K=12 THEN 1630
1610 CALL SOUND(50,110,0)
1620 GOTO 1560
1630 CALL CLEAR
1640 END
1650 CALL HCHAR(23,13,130,6)
1660 FOR I=1 TO LEN(STR$(E))
1670 CALL HCHAR(23,12+I,ASC(
SEG$(STR$(E),I,1)))
1680 NEXT I
1690 CALL HCHAR(23,25,130,6)
1700 FOR I=1 TO LEN(STR$(C))
1710 CALL HCHAR(23,24+I,ASC(
SEG$(STR$(C),I,1)))
1720 NEXT I
1730 RETURN
1740 DATA 96,000000010F1F3F7
F,97,001C7EFFFFFFFFFF,98,000
C3FFFFFFFFFFF,99,000000E0F0F
8FCFE
1750 DATA 100,7F7F3F0F03,101
,FFFFFFFFFF7E18,102,FFFFFFFF
FF71,103,FEFEFCF8F8F0E
1760 DATA 104,181818E7E71818
18,105,0003317FDF7D38,106,00
C08C9EFBFECC,107,FFFF442211F
FFFFF
1770 DATA 145,0000000100818,1
46,000024183014,147,00522C18
245224,148,442A52AD5CAA552A
1780 DATA 112,00F8E0F0F8BC9E
0E,113,0000187EFF7E,114,181E
9AFEFFFFAD52,120,387CFEFEFE7
C38
1790 DATA 108,FFFFFFFFFFFFFF
FF,109,0103070F1F3F7FFF,130,
42521084A4200909,138,0022550
8
1800 DATA 9,16,8,10,2,8,11,9
,8,12,11,8,13,13,4,14,5,8,1,
2,8,15,2,8,4,2,11
1810 DATA 12,16,150,0
1820 DATA 2,3,96,2,4,97,2,5,
98,2,6,99,2,30,120,3,3,100,3
,4,101,3,5,102,3,6,103,3,24,
96,3,25,97
1830 DATA 3,26,98,3,27,99,4,
14,96,4,15,97,4,16,98,4,17,9
9,4,24,100,4,25,101,4,26,102
,4,27,103
1840 DATA 5,14,100,5,15,101,
5,16,102,5,17,103,19,30,109,
19,31,108,19,32,108,20,29,10
9
1850 DATA 20,30,108,20,31,10
8,20,32,108,23,6,80,23,7,85,
23,8,78,21,29,108,21,30,108,
21,31,108,21,32,108
```

```
1860 DATA 23,9,75,23,10,84,2
3,11,69,23,12,58,23,20,90,23
,21,69,23,22,73,23,23,84,23,
24,58
1870 DATA 33,106,10,1,10,2,1
0,3,10,4,11,5,12,6,13,7,14,8
,15,9,15,10,15,11,15,12,15,1
3,16,14
1880 DATA 16,15,17,16,17,17,
17,18,18,19,18,20,18,21,19,2
2,19,23,20,24,20,25,20,26
1890 DATA 20,27,19,28,19,29,
18,29,17,30,16,31,16,32
1900 DATA 34,105,5,32,6,31,7
,30,8,29,9,29,9,28,9,27,9,26
,10,25,10,24,10,23,9,22,10,2
1,11,20,10,19
1910 DATA 9,18,10,17,11,17,1
1,16,10,15,9,14,8,13,9,12,10
,11,10,10,10,9
1920 DATA 11,8,12,7,11,6,12,
5,13,4,12,3,11,2,10,1
1930 DATA 15,138,4,3,5,4,6,5
,7,6,8,6,8,7,7,8,6,9,7,10,6,
11,5,12,4,12,3,13,2,14,3,15
1940 DATA 14,138,5,26,6,25,7
,24,7,23,8,22,7,21,8,20,7,19
,6,19,5,20,4,19,3,18,2,17,3,
16
1950 DATA 41,113,1,20,2,21,3
,22,4,21,5,20,6,20,7,19,7,18
,8,18,9,18,10,19,11,18,12,17
,13,16
1960 DATA 14,17,15,18,16,17,
17,18,17,19,16,20,16,21,15,2
2,14,23,14,24,14,25,13,26,14
,27,13,28,12,29
1970 DATA 13,30,14,30,15,29,
16,28,15,27,14,28,13,29,12,3
0,11,29,10,30,10,31,9,32
1980 DATA 55,113,1,19,2,19,3
,18,3,17,6,15,7,14,8,14,9,13
,10,14,11,15,12,14,13,13,14,
12,15,12
1990 DATA 16,11,17,12,16,12,
15,11,14,10,13,9,12,8,13,7,1
4,6,15,6,16,7,17,8,18,7,19,6
,18,5,17,4
2000 DATA 16,3,16,3,17,4,18,
5,19,6,18,6,17,6,16,7,15,8,1
4,9,13,10,13,10,14,9,13,10,1
2,9,11,8,11,7
2010 DATA 10,6,10,7,9,6,8,5,
8,4,8,2,9,2,10,1
2020 DATA 27,114,0,1,20,1
2030 DATA 13,112,-1,-2,19,28
2040 DATA 13,-7,2,800,145,14
5,146,146,147,147,148,148,14
7,146,146,145,32
2050 DATA 10,-6,3,650,146,14
6,147,147,148,146,146,145,14
5,32
2060 DATA 15,-7,2,650,145,14
5,146,146,147,147,147,148,14
8,147,147,146,145,145,32
2070 DATA 13,-7,2,950,145,14
5,146,146,147,147,148,148,14
8,147,146,145,32
2080 DATA 10,-5,3,250,145,14
6,146,147,147,148,148,147,14
6,32
2090 DATA GAME OVER,9,12,PRE
SS REDO OR BACK,11,7,PROC'D
= ENDE,13,10,RECORD IST :,17
,7
```

s

```
100 REM ******************
*
110 REM * P A C - M A C
*
120 REM * COPYRIGHT BY
*
130 REM * JOERN HILGER
*
140 REM * BEARBEITET VOM
*
150 REM * TEAM DES
*
160 REM * AKTUELL VERLAGES
*
170 REM * MUENCHEN
*
180 REM *BENOETIGTE GERAETE
*
190 REM * TI 99/4A KONSOLE
*
200 REM * JOYSTICKS
*
210 REM * DSK OD. KASSETTE
*
220 REM *
*
230 REM * SPEICHERBELEGUNG
*
240 REM * 4320 BYTES
*
250 REM *
*
260 REM ******************
*
270 CALL CLEAR
280 CALL SCREEN(2)
290 PRINT "..........PAC-MAC
":::::::::::::
300 PRINT "..WRITTEN BY JOER
N HILGER":
310 PRINT "..COPYRIGHT BY J.
C.HILGER"
320 PRINT
330 PRINT "...START MIT FEUE
RKNOPF"
340 CALL SCREEN(16)
350 CALL KEY(1,QR,QT)
360 CALL SOUND(500,110,2,-1,
0)
370 CALL SOUND(500,310,0,-1,
0)
380 IF QT=1 THEN 32767 ELSE
350
390 L=3
400 CALL CLEAR
410 CALL COLOR(3,2,11)
420 CALL COLOR(4,2,11)
430 CALL SCREEN(2)
440 FOR LKJ=9 TO 12
450 CALL COLOR(LKJ,16,2)
460 NEXT LKJ
470 FOR JHG=1 TO 8
480 CALL COLOR(JHG,2,2)
490 NEXT JHG
500 PRINT "CCCCCCCCCACCCCCC
ACCCCCCCCC".
510 PRINT "HAAACACAACACAAAAC
ACAACACAAAH"
520 PRINT "CCCCCACCCCCCCCCC
CCCCCACCCCC"
530 PRINT "AACAAACAAAACAAAAC
AAAACAAACAA"
540 PRINT "CCCCCCCCCACCCCCC
ACCCCCCCCC"
550 PRINT "CAAACACAACACAAAAC
ACAACACAAAC"
560 PRINT "CCCCCACCCCACACCAC
ACCCCACCCCC"
570 PRINT "CCCCCACCCCCCACCAC
CCCCCACCCCC"
580 PRINT "AAACAACAAAACAAAAC
AAAACAACAAA"
590 PRINT "CCCCCCCCCACCCCCC
ACCCCCCCCC"
600 PRINT "CAAACACAACACCCCCC
ACAACACAAAC"
610 PRINT "CCCCCACCCCACAAAAC
ACCCCACCCCC"
620 PRINT "CCCCCACCCCCCCCCC
CCCCCACCCCC"
630 PRINT "AAACAACAAAACAAAAC
AAAACAACAAA"
640 PRINT "CCCCCCCCCACCCCCC
ACCCCCCCCC"
650 PRINT "HAAACACAACACCCCCC
ACAACACAAAH"
660 PRINT "CCCCCACCCCACAAAAC
ACCCCACCCCC"
670 PRINT "CCCCCACCCCCCCCCC
CCCCCACCCCC"
680 PRINT :::
690 CALL HCHAR(1,1,65,32)
700 CALL HCHAR(2,1,65,32)
710 CALL HCHAR(21,1,65,32)
720 CALL HCHAR(22,1,65,32)
730 CALL VCHAR(1,1,65,22)
740 CALL VCHAR(1,2,65,22)
750 CALL VCHAR(1,31,65,22)
760 CALL VCHAR(1,32,65,22)
770 CALL CHAR(65,"FFFFFFFFFF
FFFFFF")
780 CALL CHAR(72,"007E7E7E7E
7E7E00")
790 CALL CHAR(66,"")
800 CALL HCHAR(9,16,66,2)
```

```
810 CALL HCHAR(10,16,66,2)
820 CALL CHAR(67,"0000001818
000000")
830 CALL COLOR(5,11,14)
840 CALL COLOR(6,13,14)
850 CALL COLOR(7,2,14)
860 E$="307C6EF8FE7C7C30"
870 F$="0C3E761F7F3E3E0C"
880 G$=E$
890 A=13
900 B=16
910 CALL SOUND(2500,300,9,-4
,0)
920 K=INT(22*RND)+1
930 Y=INT(32*RND)+1
940 CALL HCHAR(K,Y,65)
950 CALL CHAR(80,G$)
960 CALL HCHAR(A,B,80)
970 CALL KEY(1,CX,CY)
980 IF CY=1 THEN 1450
990 CALL JOYST(1,X,Y)
1000 IF X=-4 THEN 1010 ELSE
1030
1010 G$=F$
1020 GOTO 1050
1030 IF X=4 THEN 1040 ELSE 1
050
1040 G$=E$
1050 X=X/4
1060 Y=Y/-4
1070 C=A+Y
1080 D=B+X
1090 CALL GCHAR(C,D,W)
1100 CALL HCHAR(A,B,66)
1110 IF W=65 THEN 890
1120 IF W=67 THEN 1130 ELSE
1150
1130 CALL SOUND(100,300,3,-3
,5)
1140 P=P+10
1150 IF W=72 THEN 1200 ELSE
1160
1160 A=A+Y
1170 B=B+X
1180 CALL SOUND(100,-3,3)
1190 GOTO 920
1200 FOR J=1 TO 40
1210 CALL JOYST(1;X,Y)
1220 X=X/4
1230 Y=Y/-4
1240 IF X=1 THEN 1250 ELSE 1
270
1250 G$=E$
1260 GOTO 1290
1270 IF X=-1 THEN 1280 ELSE
1290
1280 G$=F$
1290 CALL HCHAR(A.B.66)
1300 A=A+Y
1310 B=B+X
1320 IF A<3 THEN 1330 ELSE 1
340
1330 A=3
1340 IF A>20 THEN 1350 ELSE
1360
1350 A=20
1360 IF B<3 THEN 1370 ELSE 1
380
1370 B=3
1380 IF B>30 THEN 1390 ELSE
1400
1390 B=30
1400 CALL SOUND(100,110,0,-3
,2,600,6)
1410 CALL HCHAR(A,B,80)
1420 CALL CHAR(80,G$)
1430 NEXT J
1440 GOTO 920
1450 FOR VX=1 TO 22
1460 FOR VY=1 TO 32
1470 CALL GCHAR(VX,VY,VB)
1480 CALL SOUND(100,110,2,23
4,5,500,3,-8,4)
1490 IF VB=67 THEN 1580
1500 NEXT VY
1510 NEXT VX
1520 PRINT ".......PPP super
 PPP"
1530 P=P+1000
1540 FOR KL=200 TO 500 STEP
100
1550 CALL SOUND(1000,KL,2)
1560 NEXT KL
1570 GOTO 400
1580 L=L-1
1590 PRINT ".......PPP schad
e PPP"
1600 FOR JK=500 TO 200 STEP
-100
1610 CALL SOUND(1000,JK,2)
1620 NEXT JK
1630 IF L=0 THEN 1640 ELSE 4
00
1640 CALL SCREEN(16)
1650 PRINT "sie erreichten";
P;"punkte ."
1660 FOR HGF=9 TO 12
1670 CALL COLOR(HGF,16,2)
1680 NEXT HGF
1690 CALL COLOR(3,2,16)
1700 CALL COLOR(4,2,16)
1710 PRINT
1720 INPUT "noch ein spiel";
FG$
1730 IF FG$="j" THEN 32767
1740 END
```

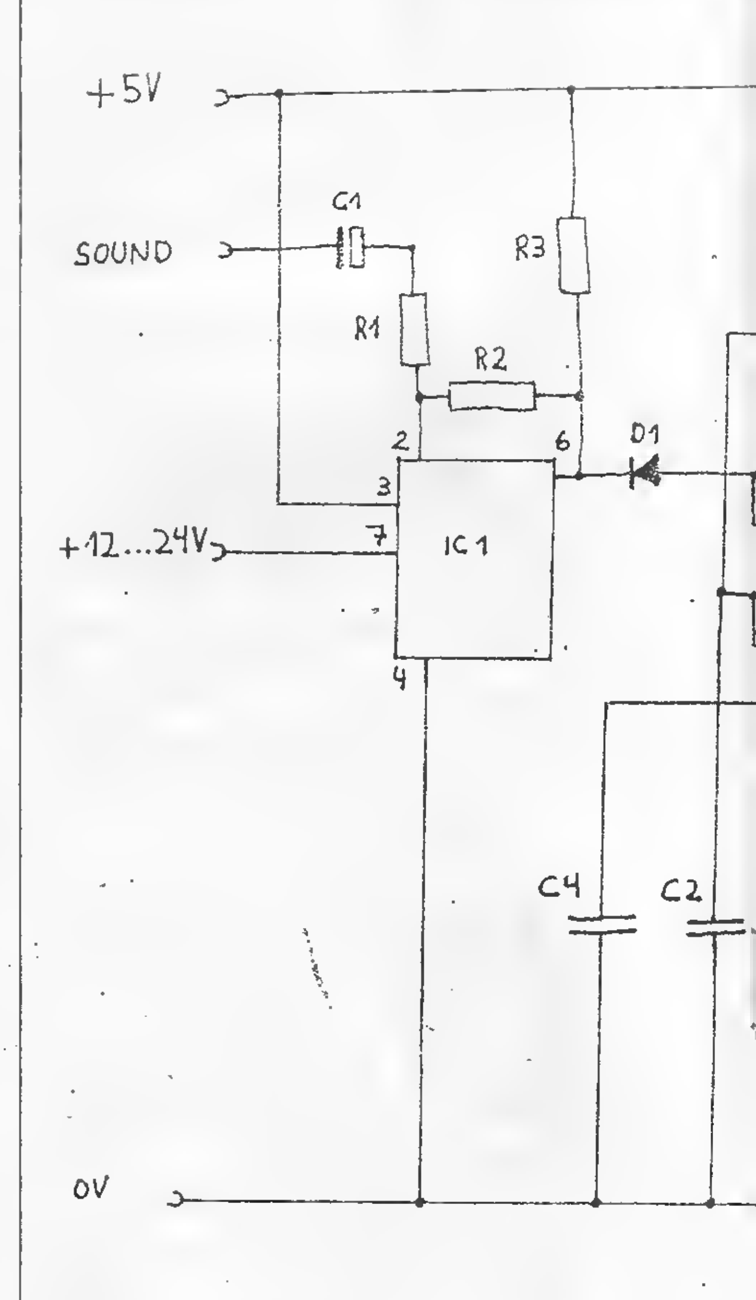
SCHALTMODUL FÜR DIE TI-K

Wie kann man mit dem TI 99 externe Geräte – etwa solche aus Mutters Küche – „fernsteuern"? Nun, man erwerbe eine Speichererweiterung und schließe das in einer vergangenen Ausgabe der „TI-Revue" beschriebene Ausgabemodul an... Für alle diejenigen Leser, welche (noch) nicht stolze Besitzer von 32 oder mehr Kbyte Prozessorram sind, folgt hier die Beschreibung einer wesentlich preiswerteren Lösungsmöglichkeit.

Leider stellt sich bei der Entwicklung irgendwelcher Peripherie für den Texas ein grundsätzliches Problem: Der Rechner verfügt – dies gilt zumindest für die reine Konsole – über keine Möglichkeit, den Datenbus gezielt anzusprechen. Für unsere Zwecke – gemeint ist der Aufbau einer „Schnittstelle" für alle Ausbaustufen – fällt der Userport des Computers somit aus. Eine Lösung unter Benutzung des Kassetten-Interfaces wurde aus naheliegenden Gründen ebenfalls verworfen. TI-Basic verlangt ja bei Benutzung eines Recorders eine intensive Rückkopplung mit dem User – anstatt mehrmalig die „Enter"-Taste zu drücken, könnten wir das zu steuernde Gerät besser mit der Hand einschalten.
Was letztlich bleibt, ist der „Sound"-Ausgang an der Videobuchse des Computers. Die Vorteile liegen auf der Hand: kein Eingriff in die Konsole notwendig, wenig Hardware und anstatt teurer Computerstecker ein billiger DIN-Stöpsel.
Wie aber kann man mit den Soundgeneratoren etwas aus- oder einschalten? Bei einem „Kanal" ist das kein Problem: Ton da – Gerät an; Ton weg – Gerät aus. Was aber, wenn man mehrere Geräte unabhängig voneinander schalten will? Dann verfährt man mit dem Tonausgang ganz einfach wie mit einer „großen" seriellen Schnittstelle: die Bits werden hübsch hintereinander hinausgejagt und im Empfänger wieder in parallel anliegende Daten verwandelt. Nach ein paar Nachmittagen im Labor war es dann vollbracht: ein 8-Bit Ausgabe Modul für TI-Basic zum Taschengeldpreis!

WIE MAN'S BAUT

Grundsätzlich benötigt man nur den linken Teil der Schaltung aus Bild 1 – also bis zur gedachten vertikalen Trennlinie A...B im Schaltplan. IC 1 – ein Operationsverstärker – sorgt für den zur Ansteuerung der TTL-ICs notwendigen Pegel des Tonsignals. Das verstärkte Signal geht über die Diode an die Eingänge von IC 2, dessen beide Timer als Monoflops geschaltet sind. Eines der Flops erzeugt das Taktsignal für die Schieberegister IC 3 und IC 4, das andere Monoflop gibt den Schieberegistern an, ob das aktuelle Datenbit hi oder lo ist.
Wie erkennt nun die Schaltung, ob ein bestimmtes Bit 1 oder 0 sein soll"? Bei einer "1" sendet der Computer zwei Piepser von je 1ms Dauer. Bei einer logischen "0" auch – der zweite Piepser ist allerdings softwaremäßig stummgeschaltet. Beim ersten Piepser gehen die Ausgänge der Flops beide auf (fast) 5 Volt. Die Schieberegister takten allerdings erst bei einer negativ gerichteten Eingangsflanke durch. Das Datenmonoflop hat nun eine kürzere Zeitkonstante als das Taktflop. Um überhaupt eine "1" am Ausgang des Schieberegisters erscheinen zu lassen, muß das Datenflop einen zweiten Piepser vom Rechner erhalten – andernfalls: wenn der zweite Ton stummgeschaltet ist, schieben IC 3 und IC 4 eine "0" durch.
Am Ausgang von IC 3 und 4 erscheinen also nacheinander unsere 8 Bits. Da sie hintereinander durch alle Ausgänge geschoben werden, dauert es ca. 1,5 Sekunden (1,2 für Ex-Basic), bis die Daten stabil anstehen.
Will man nun wirklich Haushaltsgeräte damit steuern, so reicht das auf Grund der unkritischen



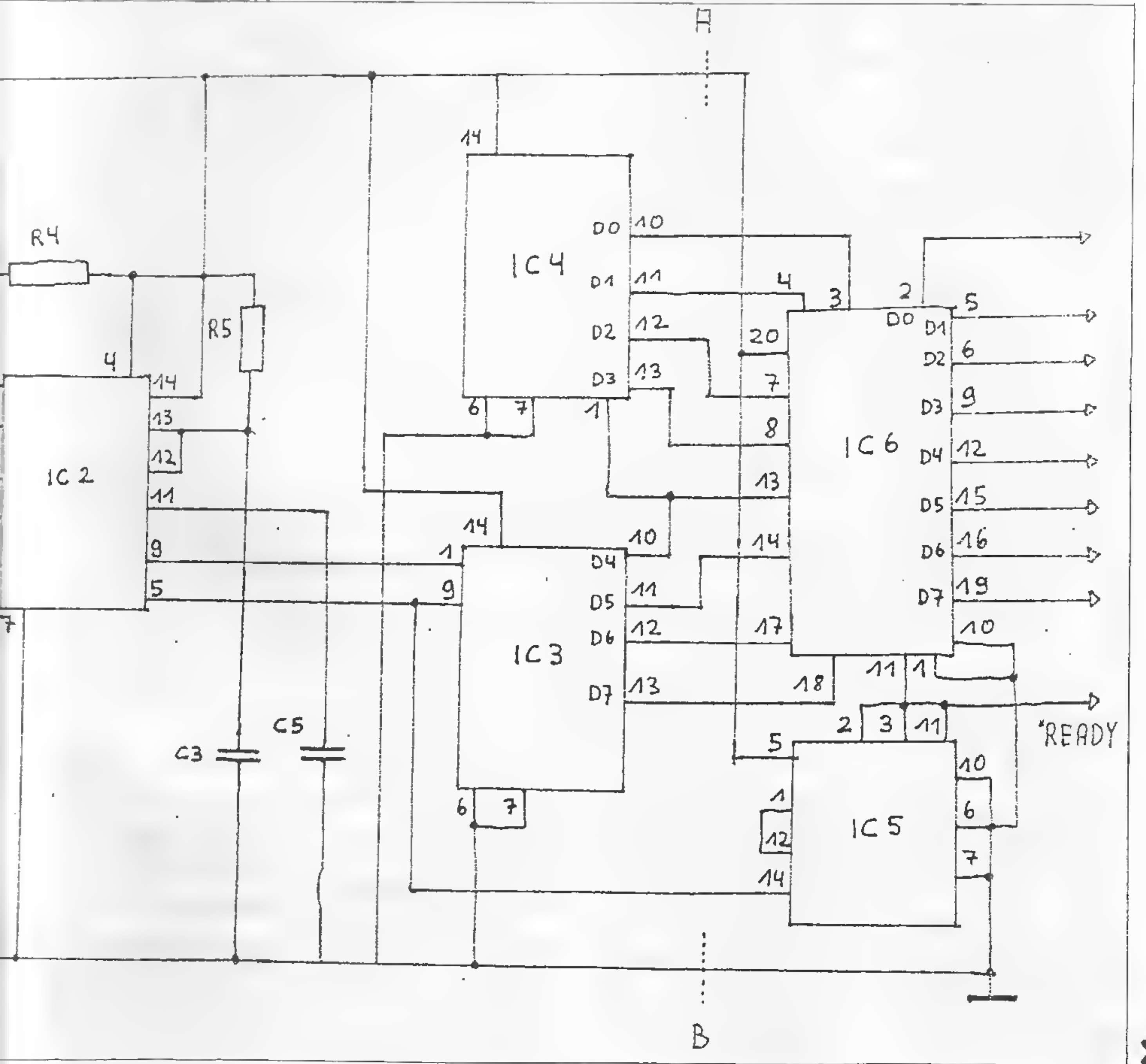
ONSOLE

Steuerzeiten vollständig aus. Andere Geräte – z.B. D/A-Wandler – benötigen einen Übernahmeimpuls bei vollständig vorhandenen Daten am Bus. Wer es braucht, kann also noch einen Dezimalzähler – IC 5 im Bild – nachschalten. Dieser ist so verdrahtet, daß er immer nach der Übertragung eines Byte einen kurzen HI-Impuls abgibt. In der Luxusversion hat man dann noch IC 6. Dabei handelt es sich um ein 8fach Speicherflipflop – die Daten werden nun erst bei beendeter Übertragung auf den Ausgang gegeben.

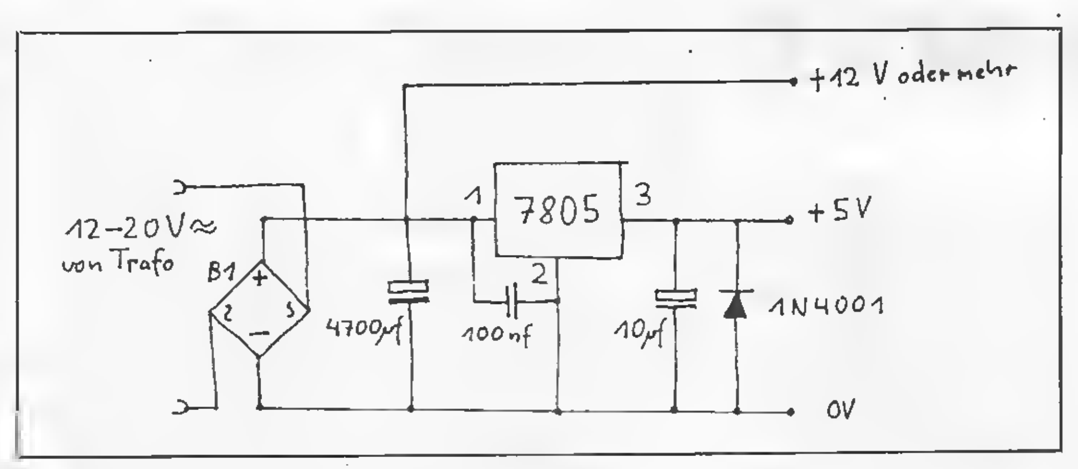
ZUM AUFBAU

Grundsätzlich sind für alle ICs Fassungen zu verwenden. Einzig winziges Problem ist die Spannungsversorgung: Um die Monoflops zu triggern, brauchen wir eine zweite Versorgungsspannung zwischen etwa 12 und 20 Volt Gleichspannung. Am unkritischsten ist es darum, ein unabhängiges Netzteil – siehe Bild 2 – aufzubauen. Damit vermeidet man unter Umständen die Einsendung der Konsole nach Freising...

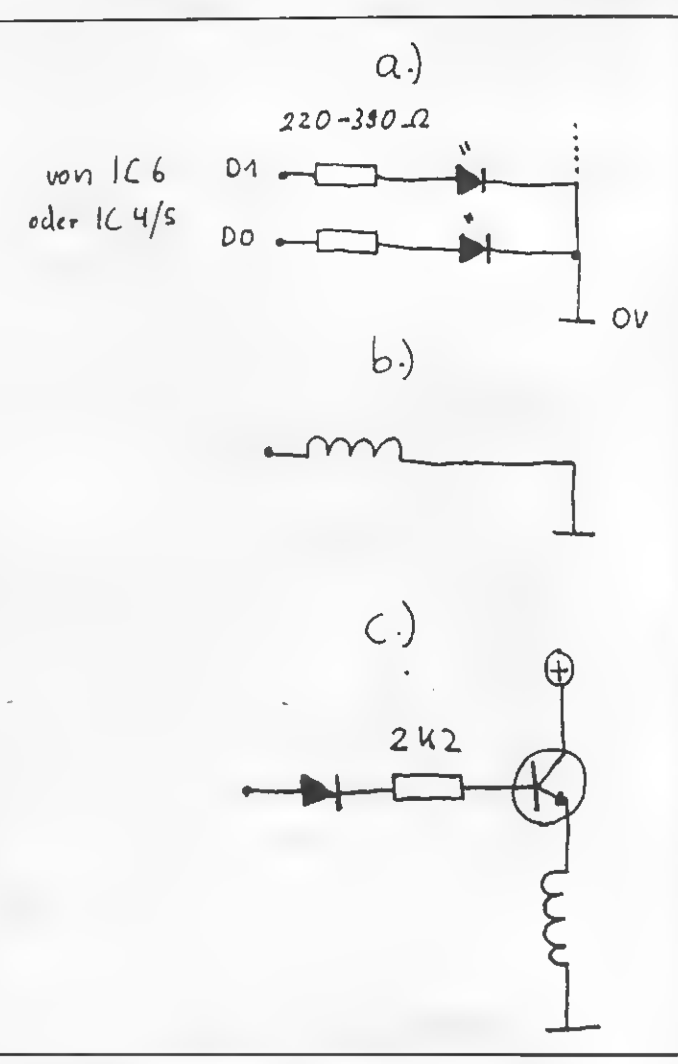
Hat man die Schaltung aufgebaut, so sollte man sich spätestens jetzt überlegen, was man eigentlich damit schalten und walten will. Es gibt z.B. im Elektronikhandel spezielle Relais, die man direkt an TTL-ICs anschließen kann. Bei dieser – ziemlich teuren – Lösung schließt man die Relais wie in Bild 3b. direkt an die Ausgänge von IC 3/4 bzw. IC 6 an. Oft kann man in Elektronikläden preiswerte Relais für hohe Ströme als Surplus-Ware bekommen. Man muß allerdings eine Treiberstufe nach Bild 3 c. vorschalten. Die Diode ist eine 1n4148, der Transistor ein BC 107 oder für höhere Ströme in der Relaisspule ein Darlington aus der BD...Reihe. Für den Anfang genügt uns aber der Anschluß von LEDs nach Bild 3 a. So kann man nämlich die

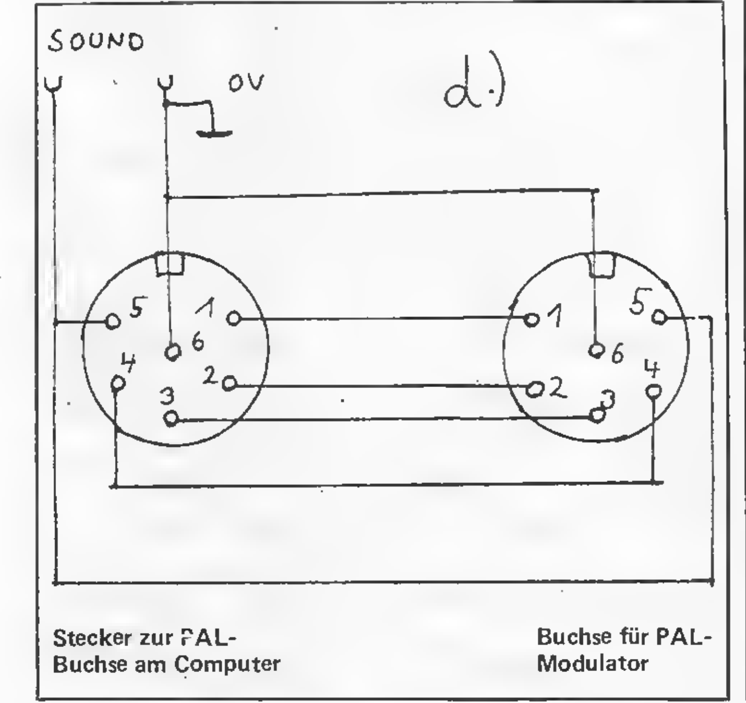


Schnittstelle recht einfach prüfen: Wenn wir die ICs in die Fassungen gesteckt und die Versorgung eingeschaltet haben, so müßten wir durch Berührung des Einganges der Schaltung die LEDs zum Flackern kriegen. Geht's nicht, dann haben wir etwas falsch gemacht. Geht's erstaunlicherweise doch, dann können wir uns überlegen, wo wir den Eingang der Schaltung anschließen. Das geht zum Beispiel am Kopfhörerausgang – insoweit vorhanden – des Monitors. Eleganter ist der Einkauf zweier handelsüblichen DIN-AV-Buchsen, welche wir dann gemäß Bild 3 d. miteinander verbinden. In die eine Buchse kommt wie gewohnt der Modulator, in die andere ein – handelsübliches – Videokabel zum Computer.



Zeile 10 ist notwendig, um den Input-Ton zu unterdrücken. Beim Programmablauf geben wir auf den Input-Befehl in Zeile 20 einen Binärstring – etwa "01000100" – ein. Die eigentliche Ansteuerung der Schnittstelle wird in den Zeilen 30–70 vorgenommen. Hier wird hintereinander jedes





Stecker zur PAL-Buchse am Computer

Buchse für PAL-Modulator

Jetzt wird alles eingeschaltet und folgendes Programm eingegeben:

```
10 CALL SOUND (500,880,
   30)
20 INPUT BIN$
30 FOR I=1 to 8
40 CALL SOUND (1,880,0)
50 CALL SOUND (1,880,30-
   VAL(SEG$(BIN$,I,1))*30)
60 CALL SOUND (10,880,30)
70 NEXT I
80 GOTO 10
```

Zeichen des Binärstrings überprüft und gemäß der Formel 30-Val(SegS(Bin$,i,1))*30) bei einer "1" ein Ton und bei einer "0" eine Pause gesendet. In unserem Beispiel müßte also die zweite und die sechste LED aufleuchten. Dieser Wert bleibt an den Ausgängen von IC 3 und 4 bzw. IC 6 erhalten, bis wir einen neuen Wert senden.

Stückliste zum Schaltplan Bild 1

IC 1 CA 3140 (Operationsverstärker)
IC 2 NE 556 (Doppeltimer)
IC 3, 4 2mal 74 LS 95 (4-Bit-Schieberegister)
C 1=1 uF 40 Volt Elko
C 2=330 nF
C 3=220 nF
C 4,5=47 nF
R 1=10 kiloohm
R 2=150 kiloohm
R 3=2k2
R 4,5=270 kiloohm
D 1=1N4148 o.ä.

Stückliste zum Netzteil

B 1= Brückengleichrichter 1,5 Amp
Trafo (siehe Text) mit Wechselspannung sek. zwischen 9 und 15 Volt, mindestens 500 mA

ZU GUTER LETZT...

...noch einige Anwendungsbeispiele und Tips. Zunächst zwei kurze Routinen zur Umwandlung von Dezimalzahlen in Binärstrings und umgekehrt:

```
 90  REM DEZ-BIN...A ist
     die umzuwandelnde Zahl
100  BIN$=" "
110  FOR I=7 TO 0 STEP-1
120  N=-(A-2↑I>=0)
130  BIN$=BIN$&STRS(n)
140  A=A-(2↑I)* n
150  NEXT I
160  RETURN
170  REM BIN-DEZ...BIN$
     wird in die Zahl A
     gewandelt
180  A=0
190  FOR I=1 TO B
200  A=A+(2↑(B-I))*VAL
     (SEG$(BIN$, I,1))
210  NEXT I
220  RETURN
```

Die Routinen erweisen sich vor allem dann als nützlich, wenn man auf speicherplatzsparende Weise möglichst viele feste Ansteuerwerte in ein Array aufnehmen will. Man kann so eine Binärzahl Bin$ vom dezimalen Wert X als Integerwert speichern:

```
X$(I)=CHR$(X)
```

EINIGE BEISPIELE

Tja, dann bleiben nur noch einige Anwendungsbeispiele zu nennen: wie wär's mit einem programmierbaren Lauflicht für den Fetenkeller? Oder mit einer Alarmanlage oder

- man steuert mit dem TI eines der billig bei Elektronikversendern erhältlichen „Movits"
- oder die Autorennbahn des kleinen Bruders.

Alle genannten Beispiele betreffen Anwendungsbereiche, bei denen es nicht auf die gemächliche Geschwindigkeit der Schaltung ankommt. Aber vielleicht fällt einem der Leser ja eine „erwachsenere" Anwendung der Ausgabeeinheit ein.

Jürgen Heinzerling

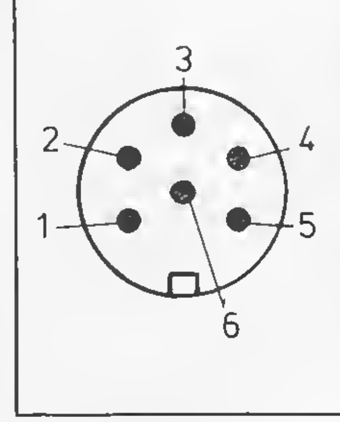
CALL LOAD ohne Risiko

Oftmals hört man abenteuerliche Geschichten, wonach verschiedene Programme oder Kopierschutzmethoden in der Lage sein sollen, Monitore, Disksysteme und Computer zu zerstören. Beim TI 99/4A war auch schon das Gerücht zu vernehmen, daß CALL PEEK oder CALL LOAD auf bestimmte Speicherbereiche zu Defekten führen kann. Im Allgemeinen kann man sagen: Kein Programm oder Kommando kann einen Computer nachhaltig beschädigen oder zerstören. Das Schlimmste was passieren kann ist, daß der Computer abstürzt oder sich aufhängt. Er weigert sich dann, irgendein Kommando von der Tastatur anzunehmen. Um wieder Kontrolle über sein System zu erlangen, muß man die Anlage abschalten und anschließend wieder von vorne beginnen. In diesem Fall ist natürlich jegliches Programm und der gesamte Speicherinhalt verloren. Bei Programmen, bei denen man damit rechnen muß, daß der Computer abstürzt, z.B. bei den ersten Ablaufversuchen von Assembler-Programmen, sollte man das Programm auf jeden Fall vorher speichern. Beim TI 99/4A gibt es eigentlich nur eine einzige Situation, bei der längerfristig Bauteile stark belastet werden können: Bei Assembler-Programmen, wenn mehrere Betriebssysteme von Peripheriegeräten gleichzeitig eingeschaltet werden. Zu erkennen ist dieser Zustand daran, daß die Lämpchen in der P-Box von RS232-P-Code-und Disk-Controller-Karte gleichzeitig leuchten. Dann sollte die Konsole schnellstmöglich abgeschaltet werden, um Bauteile, die nicht 100%ig die Werte des Datenblattes erreichen, zu schonen. Aber dieser Fall ist selbst bei Assembler-Programmierung nur durch ganz gezielte Befehle zu erreichen, im Normalfall und ganz besonders im Basic bzw. Extended Basic besteht keinerlei Gefahr, durch CALL PEEK oder CALL LOAD etwas zu zerstören.

PIN-Belegung des Modulator-Ausganges

Die Ausgänge der Buchse, an der der PAL-Modulator angeschlossen wird, sind in einigen der alten Schaltbilder von Texas Instruments falsch angegeben. Da diese Buchse auch zum direkten Anschluß eines einfarbigen Monitors genutzt werden kann, hier die korrekten Anschlüsse:

PIN 1: +12V geregelt
PIN 2: Y-Signal des Videoausganges
PIN 3: R-Y-Signal des Videoausganges
PIN 4: B-Y-Signal des Videoausganges
PIN 5: Tonsignal
PIN 6: Masse



Zum Anschluß eines monochromen Monitors wird das Y-Signal (Video) und Masse benötigt. Besitzt der Monitor auch einen Tonverstärker, so ist dieser mit dem Tonsignal zu speisen.
Zur Vermeidung von Masseschleifen sollte dieses Kabel abgeschirmt sein, aber nur am Monitor mit Masse verbunden werden. Das Video-Signal muß über ein 75Ohm-Kabel geführt werden.
Der direkte Anschluß eines Farbmonitors mit RGB-Eingang ist leider nicht möglich, hier müssen die Signale,Y, R-Y und B-Y erst entsprechend aufbereitet werden.

Heiner Martin

AND, OR, XOR, MIN und MAX für TI-Basic

In der TI-Revue 2/84 las ich, wie die Funktionen MIN und MAX des Extended Basic im TI-Basic nachvollzogen werden können. Die dort vorgestellten, langen Routinen können aber noch weiter gekürzt werden:

MAX:
```
100 C=-B*(A<=B)
    -A*(B<A)
110 RETURN
```
und MIN:
```
100 C=-A*(A<=B)
    -B*(B<A)
110 RETURN
```

Bei der Kürze dieser Unterprogramme sollten sie aber jeweils direkt im Programm eingebaut werden, um nicht unnötigen Speicherplatz zu verbrauchen.

Nun noch zu AND, OR und XOR. Auch diese Extended Basic Funktionen können im TI-Basic einfach nachvollzogen werden:

```
100 IF A=3 AND B=7
    THEN 190
```
ergibt im TI-Basic
```
100 IF (A=3)*(B=7)
    THEN 190
```
```
100 IF A=3 OR B=7
    THEN 190
```
ist im TI-Basic
```
100 IF (A=3)+(B=7)
    THEN 190
```
```
100 IF A=3 XOR B=7
    THEN 190
```
kann man im TI-Basic schreiben als
```
100 IF (A=3) - (B=7)
    THEN 190
```

Wie man sehen kann, ist auch das TI-Basic eine mächtige Programmiersprache.

Ugo Zoppi

DER KLEINE MUNTERMACHER

Eine kleine Hardwareerweiterung macht den langsamen TI 99/4A zum Rennwagen.

Haben Sie das auch schon einmal erlebt:
Sie haben eine ganz tolle Idee für ein Programm und bemerken beim Programmieren, daß Ihr Vorhaben ein Flop werden muß, da die notwendigen Routinen den Rechner viel zu langsam machen? Und dann ärgern Sie sich darüber, weil die Assemblerausrüstung für Ihren TI viel zu teuer ist? Das ist nun vorbei.
Wir haben für Sie eine Hardwareschaltung zur Assemblerprogrammierung getestet, die zum einen sehr günstig zu kaufen ist und zum anderen nur das Extended Basic-Modul benötigt.
Damit Sie uns richtig verstehen: den Cassettenrecorder brauchen Sie auch noch, denn irgendwo müssen Ihre Programme ja gespeichert werden. Sie haben richtig gelesen: Es wird kein Diskettenlaufwerk benötigt. Und jetzt in medias res.

DER KASTEN

Die kleine Hardwareschaltung findet in einem Kunststoffkasten von 100 mm x 50 mm x 25 mm Platz und besteht, wenn Sie das Gehäuse geöffnet haben, aus sehr wenigen Bausteinen. Auf der Platine finden sich zwei Widerstände, ein Chip vom Typ HM6116P-3, ein Speicherbaustein, und ein SN74LS138N. Ein kleiner Kippschalter wurde an der Gehäusestirnseite untergebracht. Der elektronische Aufbau birgt also weder Wunder noch technische Novation. Aber sinnreiche Details. Dieser kleine Kasten zeigt an beiden Längsseiten einen Peripheriebus. Mit dem einen Stecker können Sie damit Ihre Neuerwerbung an den Systembus Ihres TI anschließen und die andere Busseite für weitere Peripherie nutzen. Falls vorhanden. Die Stecker passen übrigens sauber an den TI-Bus, da klemmt oder wakkelt nichts. Ebenfalls von sehr guter Qualität ist das Kunststoffgehäuse. Es besteht aus zwei Hälften, die sehr stabil aussehen und mit vier starken Schrauben zusammengehalten werden. Einen Sturz von der Tischplatte dürfte das Gehäuse wohl aushalten. Aber versuchen Sie bitte nicht, mit Ihren Füßen die Stabilitätsgrenze des Gehäuses zu ergründen. Die beiden Gehäusehälften werden nicht einfach stumpf aufeinander gesetzt, sondern zeigen Nut und Feder. Das zeugt von durchdachter Konstruktion. Derart verbundene Bauteile können nicht wackeln oder zueinander verrutschen. Nur die Farbgebung der Gehäuseteile läßt zu wünschen übrig. Grau in Grau ist die Devise. Oberteil in Hellgrau Unterteil in Dunkelgrau. Aber Ihr TI soll ja auch kein buntes Kunstwerk sein.

AUSSTATTUNG

Das kleine Kästchen kommt natürlich nicht alleine. In seiner Begleitung finden sich eine Kassette und ein kleines Handbuch. Klein ist etwas untertrieben, denn im Format DIN A4 läßt es sich schwer verlieren und mit 25 Seiten hat es schon recht viel zu lesen.
Im Handbuch wird nicht nur beschrieben, wie Sie Ihr kleines Zauberkästchen anschließen müssen. Sie werden in einem Kurzlehrgang in die Assemblerkürzel eingewiesen. Und zwar in alle, die auf dem TMS 9900 lauffähig sind. Und das Ganze geschieht in deutscher Sprache. Obendrein findet sich ein kleines Beispielprogramm, das Ihnen den ersten Start leicht machen soll.
Sie kommen aber in keinem Fall darum herum, sich selbst mit anderer Literatur zur Assemblerprogrammierung zu beschäftigen. Und da gibt es mittlerweile einiges auf dem Büchermarkt. Einen entsprechenden Vermerk mit Literaturhinweis finden Sie ebenfalls im Handbuch. Ein, wie wir meinen, lobenswerter Zug.
Die mitgelieferte Kassette hat dann endlich alles, was Sie sich schon lange gewünscht haben:
- Line by Line-Assembler
- Dissassembler
- Kassettenladeroutine
- Kassettenspeicherroutine

ANWENDUNG

Um Ihre Erweiterung in Betrieb nehmen zu können, genügt es, sie an den TI-Bus zu stecken. Dazu muß die Konsole ausgeschaltet sein. Der schon erwähnte Kippschalter soll in der linken Position stehen. Danach schalten Sie Ihr Gerät ein, rufen Ex-Basic auf und bringen den Kippschalter in die andere Position. Jetzt können Sie den Befehl CALL INIT eingeben. Nach Eingabe von NEW ist Ihr Rechner betriebsbereit. Wenn Sie jetzt den Line by Line-Assembler von Kassette laden und starten, können Sie sofort mit der Programmierung Ihrer Ideen beginnen. Auf Wunsch erfolgt ein Protokollausdruck über einen angeschlossenen Drucker. Der Assembler ähnelt sehr stark dem vom Mini Mem, kann aber von der Geschwindigkeit nicht mithalten und hat leider auch einige Einschränkungen, die nicht verschwiegen werden:
- nur eine Zahl pro Data-Anweisung zulässig
- außer Indizierung keine Berechnung im Operandenfeld
- Programme sind auf eine bestimmte Adresse festgelegt
- keine vorgegebenen Definitionen
- das S-Symbol als Synonym für den Wert des Programmzählers ist nicht zulässig.

Mit diesen Einschränkungen läßt es sich aber ganz gut leben, zumal Sie mit diesem Assembler ein wirkungsvolles Instrument in der Hand haben, das es auch erlaubt, direkte Manipulationen im VDP vorzunehmen. Die gesamte Handhabung des Assemblers ist problemlos zu nennen und bereitet viel Spaß, wenn Sie sich erst einmal eingearbeitet haben.
Eine weitere wertvolle Hilfe stellt der Disassembler dar, mit dem Sie z.B.: fremde Maschinenprogramme untersuchen können.
Damit Ihre Programme nicht verloren gehen, können Sie alles auf Kassette abspeichern. Sie brauchen keine schwer verständlichen Befehle einzugeben, denn die Speicherroutine sucht sich die Namenstabelle, Programmstartadresse und Länge des Programms selber.
Genauso unkompliziert funktioniert das Wiedereinlesen Ihres Programms. Da kann wirklich nichts schief gehen. Dafür sorgt die Einleseroutine.

ERKLÄRUNGEN

Die auf Kassette mitgelieferten Programme befinden sich schon recht lange auf dem Markt. Daraus macht auch das Handbuch kein Geheimnis. Es handelt sich hierbei um Ex-Basic-Programme, die erstmalig 1983 veröffentlicht wurden und bis Dato nur mit der 32 KRAM liefen. Diese Erweiterung ist zwar überall käuflich zu erwerben, kostet

aber zwischen 450 und 650 DM. Die hier vorgestellte Erweiterung beläuft sich auf nur 149 DM. Selbst das Mini Mem, bisher der günstigste Einstieg in die Welt der Maschinensprache, kann da nicht mithalten. Falls Sie überhaupt noch irgendwo ein solches Modul erwerben können.
Diese kleine Hardwareschaltung läßt uns aufatmen. Der TI ist noch lange nicht tot. Jedenfalls solange nicht, wie solch gute Hardware auf den Markt dringt und auch in großen Stückzahlen verfügbar ist. Die Produktion läuft auf vollen Touren und wer sich solch ein Gerät zulegen will, der sei an die Firma RADIX Bürotechnik in Hamburg verwiesen. Sie werden mit Sicherheit nicht enttäuscht sein. Denn diese Hardware stellt eine echte Alternative zur Speichererweiterung oder dem fast nicht mehr zu bekommenden Mini Mem dar, hat fast den gleichen Befehlssatz, ist billiger und sehr einfach zu bedienen. *Klaus Weidemann*

ONLY FOR CLUBS

HALLO TI-USER CLUBS

Hier ist Eure Seite, auf der Ihr Eure Clubaktivitäten bekannt machen könnt. Aber auch alle diejenigen, die einen Club gründen wollen, können sich hier zu Wort melden.

Beginnen wollen wir heute mit einem Aufruf des TI-Computer e.V. Aachen, der sicherlich etwas stark nach Mitgliederwerbung klingt, aber doch auch interessante Anregungen enthält:

Wir, der TI-Computer e.V. mit Sitz in Aachen, wollen allen auf dem Land oder in der Stadt alleingelassenen Usern den Kontakt anbieten. Wir haben eine Vereinszeitschrift, Arbeitskreise, die auch telefonisch und schriftlich auswärtigen Usern offenstehen, um Hard- und Softwareprobleme zu lösen und viele Informationen aus erster Hand.
Wir sind ein vom Finanzamt als gemeinnützig anerkannter und beim Amtsgericht eingetragener Verein und verstehen uns als Dachverband der TI99/4A User und User-Clubs. Wir pflegen Kontakte zu anderen Organisationen im In- und Ausland und unsere Vereinszeitschrift wird als Information an Anfragende sogar direkt von Texas Instruments Deutschland weitergegeben. Solltet Ihr Kontakt mit uns suchen, dann fordert doch gegen Beilage des Rückportos in Höhe von 1,20 DM unsere neueste Vereinszeitschrift als Information an.
Natürlich könnt Ihr auch bei uns anrufen, nämlich bei unserem Vorsitzenden Hartmut Dirks, Appartment 312, Hans-Böckler-Allee 155, 5100 Aachen, Telefon-Nr.: 0241-87 22 05.
TI-COMPUTER Club Aachen

★

Ein weiterer TI-User-Club, den wir heute vorstellen wollen, hat seinen Sitz in Ulm/Donau:
Der TI-User Club Ulm trifft sich in ca. 6wöchigem Abstand im Schulungsraum eines örtlichen Computer-Fachgeschäftes. Stand zu Beginn der Clubaktivitäten der gegenseitige Austausch von Bezugsquellen der benötigten Erweiterungen im Vordergrund, TI hatte ja gerade die Produktionseinstellung bekannt gegeben, so werden heute auf den Treffen von Programmfragen bis hin zu Hardwareentwicklungen alle Themen besprochen. Der TI-User Club Ulm ist nur regional tätig, dank der Verbindungen einiger Mitglieder sind Kontakte zu anderen Clubs vorhanden. Wer Interesse hat und im Großraum Ulm wohnt, wendet sich bitte schriftlich an die folgende Kontaktadresse. Er wird dann zum nächsten Treffen eingeladen:
Heiko Liebald, Eichenweg 7, 7914 Pfaffenhofen.

★

Langsam steigt nun auch die Anzahl der TI-User, die das notwendige Zubehör zur Datenfernübertragung besitzen. Die Fa. Jochen Gerhardt u. Bettina von Megern, Höhenstr. 74b, 4000 Düsseldorf, möchte diesen Usern helfen und eine Liste mit Namen und Telefonnummern der mit DFÜ arbeitenden TI-User erstellen. Wer in diese Liste aufgenommen werden will, kann sich an die genannte Adresse wenden.

★

Zu guter Letzt für heute noch eine gute Nachricht für alle User-Clubs: Texas Instruments USA hat im letzten Herbst ein sogenanntes Care-Package an alle dort bekannten User-Clubs vergeben, die dieses Ihren Mitgliedern kostenlos zur Verfügung stellen können. Diese Care-Package umfaßt 4 Disketten Software und beinhaltet eine Single-Step Debugger, neue Diskettenfiles für die Module Multiplan und TI-Writer (letztere sind aber hier in Deutschland uninteressant, wo noch die TI-Writer Version 2.0 auf den Markt gekommen ist) und den Quellcode des TI-Forth. Einige der hiesigen User-Clubs konnten dieses Care-Package zwischenzeitlich aus den USA erhalten. User-Clubs, die dieses Care-Package noch nicht erhalten haben, sollten sich schnellstens an Texas Instruments ggf. in die USA wenden.

WISSENSWERTES FÜR DISKETTEN-FREUNDE

Die Disk ist, wie jedermann weiß, eine feine, runde Sache. Deshalb hatte ich mir noch rechtzeitig eine TI-Peripherie-Box besorgt, einschließlich eines eingebauten Diskettenlaufwerkes. Mit der gehörigen Portion Begeisterung stürzte ich mich dann auf die Tastatur – doch schon das Einführungsprogramm auf Manual-Seite 29 der PHP 1240 war eine Enttäuschung. Auch das Disketten-Katalog-Beispiel auf Seite 39 wollte nicht laufen. Nein, das Heft enthält leider erhebliche Fehler. Und überdies reicht diese Gebrauchsanleitung für blutige Anfänger kaum aus.
Da bleiben manche Fragen offen. Eine zum Beispiel, ob man die EOF-Funktion bei der Dateibildung gesondert eingeben muß. Erst wer's ausprobiert stellt fest: sie wird automatisch generiert.

IF EOF (Dateinummer) THEN CLOSE ≠ Dateinummer

entpuppt sich damit als wunderschöne Möglichkeit, Datensätze aus einer Datei zu lesen ohne das Dateiende zu überschreiten. Will man die Sätze dabei zählen, so fügt man einfach einen Zähler "Z=Z+1" hinter EOF ein, muß diesen aber vor Beginn auf Null setzen, damit die Satzzahl stimmt. EOF vermeidet jedenfalls zuverlässig, über das Dateiende hinaus zu lesen, was ansonsten unweigerlich zu ERROR 25 und Programmabbruch führt. Was hat mich diese Fehleranzeige schon geärgert!

Dabei hatte ich zumeist „nur" vergessen, die Datei rechtzeitig mit RESTORE ≠ Dateinummer in die Ausgangslage zurück zu versetzen.

Zu den Dateien selbst bringt die Betriebsanleitung die allernötigsten Angaben. Der Unterschied in der Dateiorganisation zwischen SEQUENTIAL und RELATIVE, zwei TI-spezifischen Benennungen, wird ausreichend erklärt. SEQUENTIAL, in der Literatur auch als „seriell" bezeichnet, bedeutet bekanntlich „in einer Folge aufgezeichnet", so wie dies bei Musikstücken auf Tonbändern geschieht. Der einzelne Teil einer Datei, ein Datensatz zum Beispiel, kann bei dieser Aufzeichnungsweise nicht gesondert abgerufen werden; die Datei läuft nur als ganzes in einem Stück OUT oder IN. RELATIVE hingegen heißt nichts anderes als wahlweiser Zugriff, Random access, also beliebiger Zugriff auf jeden Datensatz oder jedes Datenfeld einzeln. OPEN-Modus, Datei- und Satztyp gehen aus dem Text der Beschreibung einigermaßen verständlich hervor. Nur bei der im OPEN-Modus nach UPDATE, INPUT usw. möglichen Zahlenangabe muß man wissen, daß diese die Anzahl der Datensätze für die Datei reserviert. Und begrenzt!
Was jedoch nicht in der Betriebsanleitung steht, ist die Tatsache, daß die Dateiadressierung im OPEN-Befehl ohneweiteres auch eine numerische Variable wie DATNR akzeptiert, die im gegebenen Fall dann mit der entsprechenden Zahl aufgefüllt wird. Dasselbe gilt für eine Stringvariable im Dateititel wie "DSK1.TITELS". Nimmt das Diskettensystem beim soundsovielten Versuch den OPEN-Befehl nicht mehr an, so ist meist nur eine klitzekleine Abweichung zum vorangegangenen OPEN-Statement schuld. Löscht man die vorhergehende Datei beim Experimentieren nicht mit dem Befehl DELETE "DSK1.BLABLABLA", erweist es sich bei mehreren Dateien als nützlich, ein CALL-Unterprogramm OPEN zu verwenden. Zum Beispiel

```
SUB OPEN
(Dateinummer)
OPEN‡Dateinummer:
"DSK1.BLABLABLA";
RELATIVE, INTERNAL.
SUBEND
```

das mit CALL OPEN (Dateinummer) aufgerufen wird. Die Variable „Dateinummer" ist dann natürlich mit der konkreten Zahl zu füttern. Dieser Vorteil spart besonders bei größeren Arbeitsprogrammen Zeilen und vermeidet überdies Fehler.
Ob nun SEQUENTIAL, FIXED oder RELATIVE, die Anzahl der verzeichneten Sätze gehört bei Dateien mit fester Satzlänge in den 0-Record unmittelbar nach dem OPEN-Statement

```
PRINT‡Dateinummer:
Datensatzzahl
```

Der Inhalt der Datensätze wird dann mit

```
FOR I=1 TO Datenzahl :: PRINT‡Dateinummer:A$,B$
:: NEXT I
```

in die Datei eingeschrieben. Ohne Schleife bewegt sich nämlich nichts. Mit dem Programmteil

```
INPUT‡Dateinummer:
Datensatzzahl :: FOR
I=1 TO Datensatzzahl
:: INPUT‡Dateinummer:A$,B$ :: NEXT I
```

können die Datensätze jederzeit wieder herausgelesen werden, wobei die Bezeichnung der Stringvariablen keine Rolle spielt. Lesen darf man also mit einer völlig anderen Variablenbezeichnung als Eingeben. Dies gilt sinngemäß natürlich auch für numerische Variable. Eine an bereits aufgezeichnete Sätze anschließende Ausgabe, z.B. "FOR I= 25 TO ENDE :: PRINT‡ Dateinummer:A$,B$ :: NEXT I" funktioniert nur bei mit APPEND eröffneten sequentiellen Dateien variabler Satzlänge. Bei einer RELATIVE-Datei muß es hingegen heißen:

```
FOR I=25 TO ENDE::
PRINT‡Dateinummer,
REC I:A$,B$::NEXT I
```

Bitte dabei das CLOSE‡ Dateinummer nicht vergessen. Mit REC kann selbstverständlich auch jeder beliebige Datensatz adressiert und herausgeholt werden. Das geht auch dann, wenn eine RELATIVE-Datei in einem Stück ohne REC aufgebaut worden ist. Die Funktion REC(Dateinummer) liefert übrigens ganz bequem auch die Nummer des nächsten Datensatzes, wobei die Sätze natürlich mit 0 beginnen. So zeigt

```
FOR I=1 TO Datensatzzahl::INPUT‡
Dateinummer:A$,B$
PRINT REC (Dateinummer);") ";A$,B$
:: NEXT I
```

die einzelnen Datensätze mit der jeweils zugehöri-

gen Satznummer auf dem Bildschirm an.
Was den zwischen den einzelnen Datenfeldern notwendigen Trennseparator "," (z.B. A$,B$) betrifft, kann man bei INTERNAL-Dateien die ganze mühsam erlernte Stellenwertigkeit der Druckzeichen für PRINT und DISPLAY, wie ";" und "," oder ":", getrost vergessen. Hier genügt eindeutig das Komma zwischen A$ und B$.
völlig anders wird dies bei Dateien im Display-Format gehandhabt.
Selbst innerhalb eines gespeicherten Strings steuern Satzzeichen die DISPLAY-Ausgabe. Um diese unerwünschte Wirkung zu vermeiden, steht im Extended BASIC der Befehl LINPUT zur Verfügung, der anstelle von INPUT in Verbindung mit ‡Dateinummer auch bei Disketten angewendet werden kann. Bei X-Basic sind übrigens auch die Zeilen bei Diskettenprogrammen mit dem Zeichen "::" ohne jede Einschränkung voll ausnutzbar.
Beim Aufbau einer Output-Datei empfiehlt sich als Kontrolle, ob der letzte Datensatz auch wirklich aufgezeichnet ist, diesen mit INPUT,REC (letzter Datensatz) zu lesen und mit dem entsprechenden Satz des Arrays im Speicher zu vergleichen.

```
INPUT‡Dateinummer,
REC (letzter Datensatz):X$
IF X$=A$(letzter Datensatz) THEN PRINT
"Aufzeichnung o.k."
```

wobei A$ die Stringvariable des Arrays sein soll.
Den Inhalt der Disketten kann man bei Bedarf aus dem Katalogeintrag wunderschön lesen.

```
OPEN‡1:"DSK1.",
INPUT, RELATIVE,
INTERNAL
INPUT‡!1:A$,J,J,K
```

bringt aus dem 0-Sektor der Diskette den Disk-Namen ans Tageslicht, wobei J-K die Anzahl der noch verfügbaren Sektoren anzeigt.

```
FOR SCHLEIFE=1 TO
127::INPUT‡1:A$,
J,J,K
IF LEN(A$)=0 THEN
CLOSE‡1::PRINT A$
:P,K::NEXT Schleife
```

holt unter A$ den Dateinamen hervor, P gibt den Dateityp an (s. Gebrauchsanleitung) und K nennt die Gesamtzahl der Sektoren der Datei. Bei mehreren aufeinanderfolgenden Dateien hat man damit die Möglichkeit die Dateinummer für das OPEN-Statement zu gewinnen, wenn man diese Nummer in den Dateinamen hineinschreibt. Falls also der String A$ z.B. "5-Datei" lautet, schneidet man mit SEG$ (A$(1,1)) die "5" heraus, wandelt mit VAL in den numerischen Wert um (Dateinummer=VAL (SEG$(A$(1,1))) und eröffnet mit der auf diese elegante Weise gewonnenen Dateinummer. Für umfangreiche Datei- und Textverarbeitungsprogramme ungemein wichtig, weil die Aufteilung der Datenmengen in Futterportionen, die der Speicher frißt, die Weiterverarbeitung in etlichen Fällen erheblich erleichtert. Aber VAL-Umwandlung nicht vergessen!
Um unliebsame Programmabstürze infolge klitzekleiner Fehlerchen bei der Arbeit mit Disketten auszuschalten, sollte man bei größeren Arbeitsprogrammen den Befehl ON ERROR GOTO... einbauen und mit CALL ERR den Fehler abfragen. Ein Beispiel für die Diskettennutzung enthält "99 Spezial" (Freising: TI 1984). Alma Peschetz, die Autorin, versteht ihre Kunst auf raffinierte Weise.
Was, das haben Sie alles schon selber herausgefunden? Ich gratuliere! Sollten Sie aber nicht, dann kann ich Ihnen vielleicht doch einige Stunden eigener, bitterer Fehler-Erfahrungen ersparen.

Peter Grimm

SERVICE

VERSTECKTE TESTROUTINEN IN TI-MODULEN

Bei der Entwicklung von Programmen hat Texas Instruments Testroutinen mit in die Programme eingebaut, um so die einwandfreie Funktion des Programms feststellen zu können.
In manchen Modulen lassen sich diese Testfunktionen durch Drücken bestimmter Tastenfolgen aufrufen.
Möchte man von dem Modul "MUNCH MAN" die Testfunktionen benutzen, so muß man sofort bei Erscheinen des Titelbildes die SHIFT-Taste drücken und dabei nacheinander die Tasten 8 3 8, oder *//* betätigen. Gleich darauf erscheint ein neues Bild und die Abfrage RND(0-2):. Hier kann eingegeben werden, welche Runde man spielen möchte. Hat man eine gewählt, so erscheint sofort darauf die zweite Abfrage, SCN(0-19). SCN steht für screen. Man hat jetzt die Wahl, zwischen den 20 verschiedenen Runden eine auszusuchen. Schließlich erscheint zuletzt noch die Abfrage; MM(1-9), wieviele "MUNCH MEN" man haben möchte; es können maximal 9 sein.
Auch läßt sich mit SHIFT 8 3 8, während das Titelbild zu sehen ist, bei dem Invadermodul der Testmodus aufrufen. Es erscheint die Frage SLOW SPEED(Y/N). Die Geschwindigkeit läßt sich hier einstellen, langsam oder schnell. Danach läßt sich wie beim "MUNCH MAN" das Anfangsbild auswählen. Es erscheint SCREEN(00-40). Hat man zwei Zahlen eingegeben, so wird das Titelbild gleich darauf wieder angezeigt.
Ebenso kann bei dem Modul "MOON MINE" verfahren werden.
Der Schwierigkeitsgrad, LEVEL(00-59), ist hier die erste Abfrage, die angezeigt wird, und nach Eingabe zweier Zahlen kann man sich aussuchen, wieviele MOON MINERS man haben möchte. Auf dem Bildschirm steht dazu MOON MINERS(1-9).
Lediglich mit dem Auslösen der SHIFT 8 Taste läßt sich bei dem Modul HOPPER der Testmodus aktivieren. Es erscheint jedoch nur eine Abfrage, ROUND(0-9). Man kann sich nun eine der 10 Runden aussuchen, mit der man beginnen oder trainieren möchte.
Bei der Eingabe der Testwerte ist der Cursor und das, was man eingibt, nicht zu sehen. Hat man einen Testmodus aktiviert, so wird dieser durch TEST, was anstelle von HIGH-SCORE erscheint, angezeigt.
Für Besitzer der hier genannten Module ist der Testmodus sicher nützlich.

P. Coates

BUNDESLIGA

Mit diesem Programm können Sie sich Ihre eigene Tabelle erstellen und auf Kassette abspeichern. Ohne Änderungen am Programm können Sie dieses auch für andere Vereine und Sportarten verwenden.
Das Programm ist sehr komfortabel menügesteuert, sodaß Sie beim Erstellen der Tabelle keine Schwierigkeiten haben werden.

R. Bitter

```
100 ! **********************
110 ! *                    *
120 ! *  Bundesligatabelle *
130 ! *    Copyright by    *
140 ! *     R. Bitter      *
150 ! *                    *
160 ! *   Bearbeitet vom   *
170 ! *      Team des      *
180 ! *  Aktuell Verlages  *
190 ! *      Muenchen      *
200 ! *                    *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole  *
230 ! *     Ext. Basic     *
240 ! *  Kassettenrekorder *
250 ! *                    *
270 ! *  Speicherbelegung  *
280 ! *     4060 Bytes     *
290 ! **********************
295 !
300 CALL CLEAR :: FOR I=1 TO
 9 :: CALL COLOR(I,2,16):: N
EXT I :: CALL SCREEN(6):: CA
LL CHAR(104,RPT$("F",16))::
CALL COLOR(10,16,16)
310 DIM V$(20),PP(20),MP(20)
,TG(20),TE(20)
320 ON WARNING NEXT :: CALL
HCHAR(1,1,104,768):: CALL CL
EAR
330 DISPLAY AT(2,6):"FUSSBAL
LBUNDESLIGA"
340 DISPLAY AT(3,1):"-------
---------------------"
350 DISPLAY AT(5,6):"BITTE W
AEHLEN SIE!"
360 DISPLAY AT(8,6):"1=>VERE
INE EINGEBEN"
370 DISPLAY AT(10,6):"2=>TAB
ELLE EINLESEN"
380 DISPLAY AT(12,6):"3=>SPI
ELTAG EINGEBEN"
390 DISPLAY AT(14,6):"4=>TAB
ELLE AUSGEBEN"
400 DISPLAY AT(16,6):"5=>TAB
ELLE EINGEBEN"
410 DISPLAY AT(18,6):"6=>TAB
ELLE SPEICHERN"
420 DISPLAY AT(20,6):"7=>PRO
GRAMMENDE"
430 DISPLAY AT(24,2):"BITTE
KENNZIFFER EINGEBEN"
440 CALL KEY(0,K,S):: IF S=0
 THEN 440
450 IF K-48<1 OR K-48>7 THEN
 440
460 ON K-48 GOTO 470,570,670
,830,1340,1580,1680
470 CALL CLEAR
480 DISPLAY AT(2,1):"BITTE A
NZAHL DER VEREINE?"
490 ACCEPT AT(2,27)BEEP:Z
500 FOR J=1 TO Z
510 DISPLAY AT(7,1):USING "#
":J :: DISPLAY AT(7,2):".VER
EINSNAME"
520 DISPLAY AT(8,1):"-------
---------------------"
530 ACCEPT AT(11,1)BEEP:A$
540 V$(J)=A$
550 NEXT J
560 GOTO 320
570 CALL CLEAR
580 DISPLAY AT(10,7):"TABELL
E EINLESEN"
590 FOR I=1 TO 300 :: NEXT I
600 OPEN #1:"CS1",SEQUENTIAL
,INTERNAL,INPUT ,FIXED 120
```

```
610 INPUT #1:Z
620 FOR J=1 TO Z
630 INPUT #1:V$(J),PP(J),MP(
J),TG(J),TE(J)
640 NEXT J
650 CLOSE #1
660 GOTO 320
670 CALL CLEAR
680 DISPLAY AT(10,6):"SPIELT
AG EINGEBEN" :: FOR I=1 TO 3
00 :: NEXT I
690 CALL CLEAR
700 FOR J=1 TO Z
710 DISPLAY AT(10,1):V$(J)
720 DISPLAY AT(11,1):"------
----------------------"

730 DISPLAY AT(23,1):"VEREIN
 SPIELFREI BITTE <W>" :: DIS
PLAY AT(24,4):"SONST BITTE L
EERTASTE"

740 CALL KEY(0,K,S):: IF S=0
 THEN 740
750 IF K=87 OR K=119 THEN 81
0 ELSE 760
760 DISPLAY AT(13,1):"TORE G
ESCHOSSEN:"
770 DISPLAY AT(15,1):"TORE E
MPFANGEN:"
780 ACCEPT AT(13,19)BEEP:S
790 ACCEPT AT(15,19)BEEP:W
800 GOSUB 1740
810 CALL CLEAR :: NEXT J
820 GOTO 320
830 CALL CLEAR
840 DISPLAY AT(10,7):"TABELL
E AUSGEBEN"
850 FOR I=1 TO 300 :: NEXT I
860 CALL CLEAR
870 FOR A=1 TO Z-1
880 K=A
890 FOR J=A+1 TO Z
900 IF PP(J)<PP(K)THEN 980
910 IF PP(J)>PP(K)THEN 970
920 IF MP(J)>MP(K)THEN 980
930 IF MP(J)<MP(K)THEN 970
940 IF (TG(J)-TE(J))<(TG(K)-
TE(K))THEN 980
950 IF (TG(J)-TE(J))>(TG(K)-
TE(K))THEN 970
960 IF TG(J)<TG(K)THEN 980
970 K=J
980 NEXT J
990 X1$=V$(A):: X2=PP(A):: X
3=MP(A):: X4=TG(A):: X5=TE(A
)
1000 V$(A)=V$(K):: PP(A)=PP(
K):: MP(A)=MP(K):: TG(A)=TG(
K):: TE(A)=TE(K)
1010 V$(K)=X1$ :: PP(K)=X2 :
: MP(K)=X3 :: TG(K)=X4 :: TE
(K)=X5
1020 NEXT A
1030 IF Z<=18 THEN B=1
1040 IF Z>18 THEN B=2
1050 DISPLAY AT(1,3):B
1060 DISPLAY AT(1,5):".BUNDE
SLIGA-TABELLE"
1070 DISPLAY AT(2,1):"------
----------------------"
1080 DISPLAY AT(3,17):"TORE.
..PUNKT"
1090 CALL HCHAR(3,31,69)
1100 FOR I=1 TO 9
1110 CALL HCHAR(I+3,2,48+I)
1120 CALL HCHAR(I+13,1,48+I)

1130 NEXT I
1140 FOR I=1 TO 10
1150 CALL HCHAR(I+12,2,47+I)
1160 NEXT I
1170 CALL HCHAR(23,1,50):: C
ALL HCHAR(23,2,48)
1180 FOR I=1 TO 20
1190 CALL VCHAR(4,3,46,20)
1200 NEXT I
1210 FOR A=1 TO Z
1220 DISPLAY AT(A+3,2)SIZE(1
4):V$(A)
1230 DISPLAY AT(A+3,17):USIN
G "##":TG(A)
```

```
1240 DISPLAY AT(A+3,19):":"
1250 DISPLAY AT(A+3,20):USING "##":TE(A)
1260 DISPLAY AT(A+3,24):USING "##":PP(A)
1270 DISPLAY AT(A+3,26):":"
1280 DISPLAY AT(A+3,27):USING "##":MP(A)
1290 NEXT A
1300 DISPLAY AT(24,1):"RUECKSPRUNG BITTE LEERTASTE"
1310 CALL KEY(0,K,S):: IF S=0 THEN 1310
1320 IF K<>32 THEN 1310
1330 GOTO 320
1340 CALL CLEAR
1350 DISPLAY AT(10,7):"TABELLE EINGEBEN"
1360 FOR I=1 TO 300 :: NEXT I
1370 CALL CLEAR
1380 FOR J=1 TO Z
1390 DISPLAY AT(4,1):V$(J)
1400 DISPLAY AT(5,1):"----------------------------"
1410 DISPLAY AT(7,1):"TORE GESCHOSSEN:"
1420 DISPLAY AT(9,1):"TORE EMPFANGEN:"
1430 DISPLAY AT(12,1):"PLUSPUNKTE:"
1440 DISPLAY AT(14,1):"MINUSPUNKTE:"
1450 ACCEPT AT(7,20)BEEP:TG(J)
1460 ACCEPT AT(9,20)BEEP:TE(J)
1470 ACCEPT AT(12,20)BEEP:PP(J)
1480 ACCEPT AT(14,20)BEEP:MP(J)
1490 DISPLAY AT(23,8):"ALLES RICHTIG?"
1500 DISPLAY AT(24,8):"JA=1...NEIN=0"
1510 CALL KEY(0,K,S):: IF S=0 THEN 1510
1520 IF K=49 THEN 1560
1530 IF K=48 THEN 1450
1540 IF K<>49 AND K<>48 THEN 1510
1550 !
1560 NEXT J
1570 GOTO 320
1580 CALL CLEAR
1590 DISPLAY AT(10,6):"TABELLE SPEICHERN"
1600 FOR I=1 TO 300 :: NEXT I
1610 OPEN #1:"CS1",SEQUENTIAL,INTERNAL,OUTPUT,FIXED 120
1620 PRINT #1:Z
1630 FOR J=1 TO Z
1640 PRINT #1:V$(J),PP(J),MP(J),TG(J),TE(J)
1650 NEXT J
1660 CLOSE #1
1670 GOTO 320
1680 CALL CLEAR
1690 DISPLAY AT(10,4):"ALLE DATEN GESICHERT?"
1700 DISPLAY AT(12,12):"<J/N>"
1710 ACCEPT AT(14,14)BEEP:B$
1720 IF B$<>"J" AND B$<>"j" THEN 320
1730 END
1740 TG(J)=TG(J)+S :: TE(J)=TE(J)+W
1750 IF W=S THEN PP(J)=PP(J)+1 :: MP(J)=MP(J)+1 :: GOTO 1780
1760 IF W>S THEN MP(J)=MP(J)+2 :: GOTO 1780
1770 PP(J)=PP(J)+2
1780 RETURN
```

DAS GETPUT-UNTERPROGRAMM

Diese Programmroutine wird benutzt, um Daten in eine Datei zu schreiben und aus ihr zu lesen. Diese Datei muß vorher mit dem Prep- und Header-Unterprogrammen definiert worden sein. Die Routine wird mittels Call in der folgenden Form aufgerufen:

1. Call G (R/W, REC, FLD, V)
2. Call G (R/W, REC, FLD, V$)
3. Call G (R/W, REC, FLD, MIS,V2)
4. Call G (R/W, REC, FLD, MIS, V2$)

wobei die Werte folgende Bedeutung haben:

R/W=Schreib-Lesecode (num. Ausd.)
REC=Datensatznummer(num.Ausd.)
FLD=Feldnummer (num. Ausdruck)
V=numerischer Wert (num. Wert)
V$=StringDaten (String Ausdruck)
MIX=Rückgabecode (num.Variable)
V2=Rückgabevariable(num. Var.)
V2$=Rückgabevariable (String-Variable).

Der Wert von R/W zeigt an, ob gelesen oder geschrieben worden ist. Die Werte sind

0=schreibe gültige Daten in das Feld
1=lies Daten aus dem Feld
2=zeige fehlende Daten aus dem Feld an.

Jeder andere Wert für R/W verursacht einen Fehler. REC bedeutet Record=Datensatznummer. Im Modul Personal Recording Keeping wird dieser Wert als Seitennummer verwendet und im Statistikmodul als „Observation". Wenn der Wert größer als die höchste mögliche Datensatznummer ist, kommt es zu einem Fehler.
Wenn mit einem Schreibbefehl auf eine höhere Datensatznummer als die bisher höchste Datensatznummer zugegriffen wird, wird diese im Header als die neue Datensatznummer zwischengespeichert. Mit einem Lesebefehl an einen nicht definierten Datensatz kommt es zu unvorhersehbaren Folgen. Deshalb ist es wichtig, die Datensätze nur mit fortlaufenden Nummern zu vergeben. Negative Werte oder Null ergeben ebenfalls unvorhersehbare Ergebnisse.
FLD bezeichnet die Feldnummer innerhalb eines Datensatzes. Jeder Datensatz hat die gleichen vordefinierten Felder. Wenn die Feldnummer größer ist als die höchste bisher definierte Feldnummer, führt dieses zu einem Fehler. Negative Feldnummern oder Null führen zu unvorhersehbaren Folgen. V und V$ sind Variable, die in Datenfiles geschrieben werden. Sie können auch mathematische Ausdrücke sein; dann wird das Ergebnis abgespeichert.

FORTSETZUNG AUS HEFT 4/85 Seiten 12 – 14!

MINI EXTENDED-BASIC IN TI-MODULEN

Wenn die Werte nicht mit der Datensatzart übereinstimmen (z.B. Strings, wo nur numerische Werte zugelassen sind) kommt es zu unvorhersehbaren Ergebnissen. Ebenfalls kommt es zu Fehlern, wenn ein Feld als dezimales Feld definiert ist. mit einer Gesamtlänge von 6, davon 2 nach Kommastellen.
Hier sind sowohl 7654 als auch 7.654 unbrauchbare Werte. 7654 hat eine Stelle zuviel auf der Seite links vom Punkt und 7.654 hat eine Stelle zuviel auf der rechten Seite vom Dezimalpunkt. Wenn fehlende Daten aus dem Datenfeld herausgesucht werden sollen, muß V bzw. V$ im Callaufruf bei den Parametern mit berücksichtigt werden. Eventuell eingegebene Werte werden jedoch ignoriert.
MIS ist eine numerische Variable, mit der ein Code bereitgestellt wird. Dieser Parameter muß bereitgestellt werden, wenn Sie aus dem Datensatz lesen wollen; wird aber nicht benötigt, wenn Sie auf den Datensatz schreiben. MIS=0 bedeutet, daß Daten gefunden wurden, MIS=1 bedeutet, daß die Daten fehlen. V2 bzw. V2$ enthalten die Variablen, wenn sie aus dem Datensatz heraus lesen. Die Datensatzart, V2 für numerische Werte, V2$ für Strings, muß mit dem Inhalt des Datenfeldes übereinstimmen. Wenn Sie versuchen, mittels V2$ Daten zu lesen, oder mittels V2$ numerische Daten zu lesen, kommt es in jedem Fall zu einem Fehler. Wenn bei einem Leseversuch keine Daten gefunden werden, wird die Rückgabevariable nicht geändert.

DAS HEADER-UNTERPROGRAMM

Dieses Unterprogramm wird benutzt. um Informationen in den Header eines Datensatzes zu schreiben oder davon zu lesen. Es wird mittels Call in einer der folgenden Formen aufgerufen:

1. Call H (R/W, Info, FLD, V)
2. Call H (R/W, Info, FLD, V$)

wobei die Werte folgende Bedeutung haben:

R/W=Vorgangscode (num. Ausdr.)
Info=Headerartnummer (num. Ausdruck)
FLD=Feldnummer (num. Ausdr.)
V=Variablenname (num. Ausdruck)
V$=Variablenname (Stringausdruck)

R/W ist der Schreib-Lesecode mit folgenden Bedeutungen:
0=Schreibinformation in Header
1=Leseinformation im Header
Info enthält einen Wert von 1 bis 14, der angibt, welche Art von Daten in den Header geschrieben bzw. gelesen werden. Die Struktur ist wie folgt aufgebaut:

1. Datensatzname 0 bis 9 Zeichen,
2. Datum (Tag) Zahl von 1 bis 31,
3. Datum (Monat) Zahl von 1 bis 12,
4. Datum (Jahr) Zahl von 0 bis 99,
5. Anzahl der Felder je Datensatz. Dieser Wert wird automatisch aktualisiert. und zwar jedesmal, wenn eine neue höchste Feldnummer definiert wird.
6. Anzahl Datensätze. Dieser Wert wird ebenfalls automatisch aktualisiert und zwar jedesmal. wenn mittels Call Getput (Call G) ein neuer „höchster" Datensatz geschrieben wird.
7. Länge des Headers in Bytes. Dieser Wert wird automatisch zugeordnet.
8. Länge jedes Datensatzes in Bytes. Dieser Wert wird automatisch zugeordnet.
9. Feldname=bis 9 Zeichen.
10. Art des Feldes:
    1=Alphazeichen
    2=ganzzahlige Werte
    3=Dezimalwerte
    4=Expotentialwerte
11. Feldlängen Alphazeichen.
    (1 bis 15) Ganzzahlwerte
    (1 bis 10) Dezimalwerte
    (2 bis 11) Expotentialfunktionen
    (8 bis 13) Länge wird automatisch zugeordnet.
12. Anzahl der Nachkommastellen je Feld. Bei Alphazeichen 0 (automatische Zuordnung). Ganzzahlwerte 0 (automatische Zuordnung). Dezimalwerte 1 bis Feldlänge minus 1. Expotentialfunktionen 0 bis 5.
13. Speicherplatz für Felder in Bytes. Dieser Wert wird automatisch zugeordnet.
14. Position des Feldes innerhalb des Datensatzes. Dieser Wert wird automatisch zugeordnet.

Hinweis: Werte 9 bis 14 werden für jedes Feld automatisch wiederholt. FLD bezeichnet die Feldnummer. Dieser Wert wird von den vorgezeichneten Punkten 1 bis 8 ignoriert, muß jedoch in der Parameterliste aufgeführt werden.
Wenn Daten in den Header geschrieben werden oder vom Header gele-

sen werden, bezeichnet V bzw. V$ die Konstante, Variable, Ausdruck bzw. String, die jeweils geschrieben oder gelesen werden. Auf die jeweils richtige Zuordnung von numerischen bzw. Stringvariablen ist zu achten. Nachstehend nun noch 2 Beispielprogramme, die die vorstehend erläuterten Unterprogrammaufrufe verwenden. Das 1. Programm wandelt ein Personalrecordkeepingfiles in 2 Standardbasicfiles um. Das erste der beiden ist der Header und das zweite sind die Daten. Das Programm geht davon aus, daß die Dateien auf eine Diskette gespeichert sind. Vor Programmausführung sind folgende 3 Befehle im Direktmodus unter Basic (Option 1) durchzuführen.

> Call Files (1)
> Call P (10000)
> New

Lassen Sie nun das erste Beispielprogramm ablaufen. Nachstehend noch eine Erläuterung für die Bedeutung jeder einzelnen Programmzeile.

100: Lädt den Datensatz in den reservierten Speicherbereich.
110: Fehlerabfrage.
0=Fehler ‡0=erfolgreich geladen.
120: eröffne Header-Datensatz.
130: Lese internen Datensatznamen
140: Lese Anzahl der Felder je Datensatz.
150: Lese Anzahl der Datensätze.
160: Schreibe Information in das Headerteil.
170: Schreibe Information auf den Bildschirm.
180: Schleife, um Felddefinitionen auszulesen.
190: Lese Feldname.
200: Lese Feldtype
210: Lese Feldlänge.
220: Erkenne Schreibweise.
230: Alphafeld: Länge=Länge plus 1.
240: Unbedingter Sprung.
250: numerische Feldlänge = Länge plus 9.
260: Lies Anzahl der Nachkommastellen.
270: Schreibe Information in das Headerfile.
280: Schreibe Information auf den Bildschirm.
290: Ende der Schleife.
300: Schließe Header-Datensatz.
310: Öffne Datenfile
320: Eröffne Schleife, um Datensätze zu lesen, um in das Datafile zu schreiben.
330: Schreibe Datensatznummer auf den Bildschirm.
340: Eröffne Schleife, um Felder jedes Datensatzes zu lesen.
350: Lies Feldtype.
360: Prüfe Feldtype.
370: numerisches Feld, schreibe Variable ins numerische Feld.
380: Prüfe Code für fehlende Daten.
390: Fehlende numerische Daten, speichere negativen unendlichen Wert. (Wie fehlende Daten im Programm behandelt werden ist einzig und allein Ihre Entscheidung.)
400: Daten gefunden, schreibe.
410: Schreibe Daten auf den Bildschirm.
420: Unbedingter Sprungbefehl.
430: Alphafeld, schreibe in die String-Variable.
440: Prüfe Code für fehlende Daten.
450: Fehlende String-Daten, speichere eine Leerstelle. (Wie fehlende Daten gehandhabt werden ist einzig allein Ihre Entscheidung.)
460: Daten gefunden, schreiben.
470: Schreibe Daten auf den Bildschirm.
480: Ende der Schleife.
490: Beende den Datensatz zu schreiben.
500: Beende Bildschirmausgabe.
510: Ende der Datensatzschleife.
520: Schließe Datensatz.
530: Stop.
540: Fehlerausgabe aus Sprungbefehl von Zeile 110.
550: Programmende.

Das Ergebnis dieses Programmes ist
1. ein Header-Datensatz mit folgendem Inhalt
a) der Dateiname, Anzahl der Felder je Datensatz und Anzahl der Datensätze, die in Satz 0 gespeichert sind,
b) die Definition (Name, Art, Länge, Nachkommastellen) der Felder in den jeweiligen Datensätzen.
2. Ein Dateifile mit den entsprechenden Datensätzen.

Beispielprogramm 2:

Das folgende Basicprogramm nimmt den Headerdatensatz und die Datadatensätze (wie z.B. vom vorstehenden Programm erzeugt) und wandelt sie in ein Personal-Record-keeping-File um, das anschließend mit der Save-Routine abgespeichert werden kann.
Folgende drei Befehle sind ebenfalls vor Programmausführung im Direktmodus einzugeben:

> Call Files (1)
> Call P (10000)
> New

Führen Sie nun das Programm aus. Für jede Programmzeile nachstehend wieder die Erläuterungen:

100: Eröffne den Headerdatensatz.
110: Lies Datensatzname, Anzahl der Felder je Datensatz und Anzahl der Datensätze.
120: Schreibe Information auf den Bildschirm.
130: Schreibe Information in den Header des Personal-Record-Keeping-Datensatzes.
140: Öffne Schleife, um sämtliche Werte in den Header des Personal-Record-Keeping zu schreiben.
150: Lese Feld, Name, Art, Länge und Anzahl der Dezimalstellen.
160: Schreibe Information auf den Bildschirm.
170: Schreibe Feld, Name in den Personal-Record-Keeping-File-Header.
180: Schreibe Feldlänge in den Header.
190: Prüfe die Art, bei Expotentialschreibweise schreibe keine Länge in den Header.
200: Schreibe Feldlänge in den Personal-Record-Keeping-Header.
210: Prüfe Art, für Buchstaben und ganzzahlige Felder, schreibe keine Nachkommastellen.
220: Schreibe Anzahl der Dezimalstellen in den Header.
230: Ende der Schleife
240: Schließe das Headerfile.
250: Eröffne das Dateifile
260: Eröffne Schleife, um Dateidatasätze zu lesen und sie in das Personal-Record-Keeping-File zu schreiben.
270: Schreibe Datensatznummer auf den Bildschirm.
280: Eröffne Schleife, um die Felder jedes Datensatzes zu lesen.
290: Lese Feldart.
300: Prüfe Feldart.
310: Numerisches Feld, schreibe ins numerische Variable
320: Schreibe Daten auf den Bildschirm.
330: Prüfe, ob diese Eingaben als fehlende Daten interpretiert werden sollen.
340: Normale Daten, schreibe diese in das Personal-Record-Keeping-File.
350: Unbedingter Sprung.
360: Stelle fehlende numerische Daten fest.
370: Unbedingter Sprungbefehl.
380: Alphanumerisches Feld, schreibe in String-Variable.
390: Schreibe Daten auf den Bildschirm.
400: Prüfe, ob Daten fehlen.
410: Normale Daten, schreibe sie in das Personal-Record-Keeping-File.
420: Unbedingter Sprungbefehl.
430: Stelle fehlende Alpanumerische Daten fest.
440: Ende dieser Schleife.
450: Beende diesen Datensatz.
460: Beende die Bildschirmausgabe.
470: Ende der Datensatzschleife.
480: Schließe das Dateifile.
490: Speichere Personal-Record-Keeping-File ab.
500: Prüfe Fehleranzeige
0=Fehler ‡0=erfolgreich.
510: Gib Fehlermeldung auf Bildschirm aus.
520: Programmende.

SOFTWARE IM TEST

THE MINE

Vertrieb: Fa. Radix/Hamburg, Disk für Ext. Basic mit Speichererweiterung.
Nach dem Laden des Spieles befindet man sich in einem von insgesamt 25 Räumen. Die Aufgabe besteht nun darin, alle schwarzen Fugen der Mauersteine weiß einzufärben. Dies klingt sehr einfach, wenn nicht in jedem der Räume andere Hindernisse zu überwinden und gefährliche Situationen zu bewältigen wären. Alle Gefahren, Hindernisse und Spielsituationen aufzuzählen würde den Rahmen dieser Beurteilung bei weitem sprengen. Um auch den Anfänger in den Genuß aller Räume kommen zu lassen. kann man von einem Raum in den anderen gelangen auch ohne die gestellte Aufgabe ganz zu erfüllen. Nach jedem Spieldurchgang oder auch auf Wunsch während des Spieles wird ein Grundriß aller Räume und der augenblickliche Standort angezeigt. Wen die hübsche Hintergrundmusik stört, der kann sie während des Spielverlaufes wahlfrei ein- oder ausschalten. Mit der Grafik hat sich der Autor sehr viel Mühe gegeben, dementsprechend ist sie auch gelungen, Wer keinen Joystick besitzt, kann dieses Spiel auch über die Tastatur steuern. Alles in allem ist dies ein hervorragendes Spiel für den TI 99/4A mit viel Komfort und Abwechslung. Viel Spaß und lange Spielfreude dürften garantiert sein bei diesem Spiel aus der Großfamilie der Leiterspiele.

Bewertung:

Spielidee	++
Grafik	+++
Motivation	++++
Gesamturteil	+++

NEUE SPIEL-MODULE

SUB-HUNT

Vertrieb: Fa. Radix/Hamburg, Disk für Ext. Basic mit Speichererweiterung.
Sub-Hunt ist ein Kriegsspiel. in dem der Spieler als U-Bootkommandant 6 feindliche Convoys vor dem Erreichen des sicheren Hafens vernichten muß. Die Convoys werden von Zerstörern und Flugzeugen begleitet, welche ihre Waffen wirkungsvoll einzusetzen wissen. Als Hilfsmittel stehen dem Spieler Seekarte, Sonar und Periskop zur Verfügung. Da der Joystick mehrfach belegt wurde, fällt die Steuerung sehr schwer und eine Eingewöhnungszeit ist unbedingt notwendig, um bei diesem Spiel überhaupt eine Chance zu haben. Die Grafik für den Innenraum des U-Bootes ist einigermaßen gelungen, doch bei den Sprites der Schiffe versagte anscheinend die Fantasie des Autors.
Zum Laden des Spieles wird die Diskettenstation, die 32K-Erweiterung und wahlweise Mini-Memory-, E/A- oder X-Basicmodul benötigt.

Bewertung:

Spielidee	++
Grafik	++
Motivation	++
Gesamturteil	++

STAR WARS

Vertrieb: Fa. RADIX/Hamburg, Disk für Ext. Basic mit Speichererweiterung. Das Spiel entspricht dem Atari-Modul „The Empire strikes back". Um dieses Spiel auch mit dem Mini-Memory Modul laden zu können, wurde auf den Sound vollkommen verzichtet.
Ein lautloser Kampf zwischen einem Jet und 14 Stahlmonstern spielt sich vor einer Gebirgslandschaft ab.
4 Jets stehen für das gesamte Spiel zur Verfügung. Mit ihrer Hilfe und einer Laserkanone müssen pro Runde 4 Stahlmonster vernichtet werden. Dies gelingt nur mit sehr gut gezielten Schüssen. Der Gegner erwidert das Feuer mit Laserkanonen und mit Lenkwaffen. Nach jeder siegreich beendeten Runde werden die Monster aggressiver und schneller. In der unteren Bildschirmmitte wurde eine Hilfszieleinrichtung generiert, auf die man sich aber nicht unbedingt verlassen sollte.
Die obere Bildschirmhälfte beinhaltet außer den Anzeigen für Punkte, Highscore, Runde, Jets und Treibstoff noch ein Sichtfenster. in dem die Kampfhandlung verkleinert wiedergegeben wird. Die Stahlmonster bewegen sich in diesem Ausschnitt langsam von links nach rechts. Erreichen sie den rechten Rand, so ist die Aufgabe nicht gelöst und das Spiel beginnt von neuem.
Daß dieses Spiel in Assembler geschrieben wurde. garantiert für eine rasante Kampfhandlung, die (vielleicht auch durch das Fehlen jeglicher Kampf- oder Fluggeräusche) auf die Dauer doch recht eintönig werden dürfte

Bewertung:

Spielidee	++
Grafik	+++
Motivation	++
Gesamturteil	++

MACHEN SIE MIT!
VERDIENEN SIE GELD MIT IHREM COMPUTER!

Haben Sie einen Commodore VC 20 oder C 64? Können Sie programmieren? In Basic oder Maschinensprache? Dann bietet CBM-Revue Ihnen die Möglichkeit, mit diesem Hobby Geld zu verdienen!

Wie? Ganz einfach. Sie senden uns die Programme, die Sie für einen Abdruck als geeignet halten, zusammen mit einer Kurzbeschreibung, aus der auch die verwendete Hardware – eventuelle Erweiterungen – benutzte Peripherie – hervorgehen muß (Schauen Sie sich dazu den Kopf unserer Programmlistings an.)

Benötigt werden: Zwei Listings des Programms sowie eine Datenkassette oder Diskette! Wenn die Redaktion sich überzeugt hat, daß dieses Programm läuft und sich zum Abdruck eignet, zahlen wir Ihnen pro Programm je nach Umfang bis zu DM 300,–! Sollten Sie keinen Drucker haben, genügt der Datenträger.

Sie erhalten Ihre Kassette selbstverständlich zurück, wenn Sie einen ausreichend frankierten Rückumschlag mit Ihrer Adresse beifügen.

Bei der Einsendung müssen Sie mit Ihrer Unterschrift garantieren, daß Sie der alleinige Inhaber der Urheber-Rechte sind! Benutzten Sie bitte anhängendes Formular! (Wir weisen darauf hin, daß auch die Redaktion amerikanische und englische Fachzeitschriften liest und „umgestaltete" Programme ziemlich schnell erkennt).

Um Ihnen die Arbeit zu erleichtern, finden Sie hier ein Formular. Sie können es ausschneiden oder fotokopieren.

PROGRAMMANGEBOT

Name des Einsenders: ______________________

Straße/Hausnr./Tel.: ______________________

Plz/Ort: ______________________

Hiermit biete ich Ihnen zum Abdruck folgende(s) Programm(e) an:

____

____

____

____

Benötigte Geräte: ______________________

____

Beigefügt ( ) Listings ( ) Kassette ( ) Diskette

Ich versichere, der alleinige Urheber des Programmes zu sein!
Hiermit ermächtige ich die Redaktion, dieses Programm abzudrucken. Sollte es in den Kassetten-Service aufgenommen werden, erhalte ich auch dafür eine entsprechende Vergütung.

____

Rechtsverbindliche Unterschrift

TI REVUE
Postfach 1107
8044 Lohhof